



TUGAS AKHIR – TE 145561

RANCANG BANGUN APLIKASI SISTEM KENDALI WARNA LED RGB TERPUSAT BERBASIS ANDROID

Anang Maruf
NRP 2214030005

Dosen Pembimbing
Suwito, ST., MT.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT – TE 145561

DESIGNING APPLICATION OF RGB LED CENTRALIZED COLOR CONTROL SYSTEM BASED ANDROID

Anang Maruf
NRP 2214030005

Advisor
Suwito, ST., MT.

COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Rancang Bangun Aplikasi Sistem Kendali Warna LED RGB Terpusat Berbasis Android**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 18 Juli 2017

Anang Maruf
NRP 2214030005

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN APLIKASI
SISTEM KENDALI WARNA LED RGB TERPUSAT
BERBASIS ANDROID**

TUGAS AKHIR


Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada

Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing


Suwito, ST., MT.

NIP. 19810105 200501 1004

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

RANCANG BANGUN APLIKASI SISTEM KENDALI WARNA LED RGB TERPUSAT BERBASIS ANDROID

Nama : Anang Maruf
Pembimbing : Suwito, ST., MT.

ABSTRAK

Perkembangan teknologi *smartphone* Android dewasa ini tidak hanya digunakan untuk kebutuhan pesan singkat dan komunikasi suara saja, namun kemampuan *smartphone* sudah dapat mengotomasi segala sesuatu dengan aplikasi seperti yang dilakukan oleh komputer.

Pada tugas akhir ini dibuat aplikasi kendali LED RGB melalui platform *app inventor* untuk mengatur warna LED RGB sesuai keinginan pengguna. Sistem kendali warna LED RGB menggunakan mikrokontroler arduino sebagai pusat kendali untuk membangkitkan keluaran *Pulse Width Modulation* (PWM). Aplikasi android memberikan instruksi kepada mikrokontroler arduino melalui komunikasi via *Bluetooth* berupa pengiriman data dalam bentuk *byte*.

Hasil perancangan aplikasi adalah sistem pengendalian warna berdasarkan pada data palet warna yang diperoleh dalam aplikasi. Disamping itu, pengguna dapat menggunakan fitur *auto-fading* untuk menghasilkan animasi perubahan warna secara otomatis. Dapat menggunakan fitur *music synched* untuk membuat cahaya LED RGB dinamis terhadap perubahan suara di area sekitar. Serta dapat menampilkan warna LED RGB sesuai dengan objek yang direkam oleh kamera berdasarkan sistem *image processing*.

Hasil pengujian yang didapatkan proses pengiriman data *byte* dari android ke arduino melalui modul *Bluetooth* bekerja optimal dengan tingkat akurasi 95.21%, namun terbatas jarak maksimal 9 - 10 meter.

Kata Kunci : aplikasi, sistem kendali, LED RGB, *app inventor*, arduino, pengolahan citra.

-----Halaman ini sengaja dikosongkan-----

DESIGNING APPLICATION OF RGB LED CENTRALIZED CONTROL SYSTEM BASED ANDROID

Name : Anang Maruf
Advisor : Suwito, ST., MT.

ABSTRACT

The development of Android smartphone technology today is not only used for sending short messages and voice communications, but the abilities of the smartphone can already automate everything with the application as the computer done.

This final project created RGB LED control application through app inventor platform to set the color of RGB LED as user's wish. The RGB LED color control system used an arduino microcontroller as a controller to generate Pulse Width Modulation (PWM). The android application gives instructions to the arduino through Bluetooth communication with sent data byte form.

This application produces color and light intensity control system on 3 RGB LED. The color changed based on the color pallete data obtained on the application. In addition, users can use auto-fading feature to generate animated color changes automatically. Can use the music synched feature to create a dynamic RGB LED to the sound's change in the around area. And can display the RGB LED color in appropriate with the object which recorded by the camera based on image processing system.

The results of the test is data byte transfer process from android to arduino through the Bluetooth module works optimally with percentage of accuracy 95.21%, but limited the maximum distance of 9-10 meters.

Keywords : *application, control system, LED RGB, app inventor, arduino, image processing*

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Bidang Studi Komputer Kontrol, Departemen Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

RANCANG BANGUN APLIKASI SISTEM KENDALI WARNA LED RGB TERPUSAT BERBASIS ANDROID

Dalam Tugas Akhir ini dirancang aplikasi berbasis android pada sistem kendali warna LED RGB yang memberikan instruksi kepada mikrokontroler arduino melalui komunikasi via *Bluetooth*.

Dengan terselesaikannya Tugas Akhir ini, Penulis menyampaikan terima kasih yang sebesar - besarnya kepada kedua orang tua yang senantiasa mendoakan dan memberikan dukungan dengan tulus tiada henti. Bapak Suwito, ST., MT. selaku dosen pembimbing. Bapak Achmad Syaichu Buchori, S.Ag. dan Ibu Ijemiatik selaku guru Madrasah Al Anwar yang selalu memberikan doa dan restu. Al Habib Idrus bin Muhammad Alaydrus yang senantiasa memberikan doa dan restu. Teman – teman UKM Cinta Rebana, Komunitas Remaja Muslim, dan PAC IPNU Mulyorejo yang selalu memberikan semangat dan dukungan. Serta semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 18 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan.....	3
1.7 Relevansi	4
BAB II TEORI DASAR	5
2.1 App Inventor	5
2.2 Sensor Suara KY-038	6
2.3 Modul Bluetooth HC-05	7
2.4 Mikrokontroler Arduino Mega	9
2.5 Transistor Mosfet	10
2.6 Light Emitting Diode RGB	10
2.7 Raspberry Pi 3	11
2.8 Open CV	14
BAB III PERANCANGAN SISTEM	15
3.1 Kerangka Konseptual	15
3.2 Blok Fungsional Sistem	16
3.3 Konsep Perancangan Perangkat Lunak	17
3.4 Konfigurasi Pin Mikrokontroler.....	18

3.5 Perancangan Aplikasi Android dengan APP Inventor	19
3.6 Pemrograman Kode Blok Aplikasi	26
3.7 Perancangan Program Arduino	33
3.7.1 Program Pembangkit PWM	33
3.7.2 Program Auto Fading.....	35
3.7.3 Program Music Synched	35
3.8 Perancangan Sistem Image Processing	37
3.8.1 Blok Diagram Sistem Image Processing	37
3.8.2 Diagram Alir Program Image Processing	38
3.8.3 Komunikasi Raspberry Pi dengan Arduino.....	40
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....	41
4.1 Implementasi Antarmuka Aplikasi Android	41
4.1.1 Halaman Login.....	41
4.1.2 Halaman Menu.....	42
4.1.3 Halaman LED	42
4.1.4 Halaman Animasi.....	43
4.2 Pengujian Antarmuka Aplikasi Android.....	43
4.2.1 Pengujian Antarmuka Login	43
4.2.2 Pengujian Antarmuka Menu	45
4.2.3 Pengujian Pairing Bluetooth Device	46
4.2.4 Pengujian Kendali Switch LED	47
4.2.5 Pengujian Kendali Animasi	48
4.3 Pengujian Sistem Image Processing	49
4.4 Pengujian Keseluruhan Sistem	51
BAB V PENUTUP	59
DAFTAR PUSTAKA	61
LAMPIRAN A TABEL HASIL PENGUJIAN	63
LAMPIRAN B PROGRAM KODE BLOK APP INVENTOR.....	79
LAMPIRAN C PROGRAM ARDUINO.....	91
LAMPIRAN D PROGRAM RASPBERRY PI (PYTHON)	105
LAMPIRAN E DATASHEET.....	107
LAMPIRAN F DOKUMENTASI PENGUJIAN	127
DAFTAR RIWAYAT HIDUP	129

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Tampilan Halaman Desain Aplikasi App Inventor	5
Gambar 2.2 Sensor Mikrofon KY-038.....	7
Gambar 2.3 Modul Bluetooth HC-05.....	7
Gambar 2.4 Konfigurasi Pin HC-05.....	8
Gambar 2.5 Bluetooth-to-Serial-Module	8
Gambar 2.6 Mikrokontroler ATmega 2560	9
Gambar 2.7 Simbol MOSFET Channel N	10
Gambar 2.8 RGB LED Strip	11
Gambar 2.9 Raspberry Pi 3 Model B	12
Gambar 2.10 Pin GPIO Raspberry Pi 3.....	13
Gambar 3.1 Kerangka Konseptual	15
Gambar 3.2 Blok Diagram Fungsional Sistem.....	16
Gambar 3.3 Sketsa halaman login software android	20
Gambar 3.4 Sketsa halaman menu software android	21
Gambar 3.5 Sketsa halaman led software android	22
Gambar 3.6 Sketsa halaman animasi software android	23
Gambar 3.7 Sketsa halaman mode otomatis android	24
Gambar 3.8 Sketsa halaman mode auto-fading android.....	25
Gambar 3.9 Flowdiagram halaman login aplikasi android	26
Gambar 3.10 Source-Code halaman menu aplikasi android	27
Gambar 3.11 Flowdiagram pemilihan menu aplikasi android	27
Gambar 3.12 Source-Code program color picker aplikasi android	28
Gambar 3.13 Source-Code program LED Switch aplikasi android	29
Gambar 3.14 Source-Code program animasi aplikasi android.....	30
Gambar 3.15 Source-Code program music synched aplikasi android....	31
Gambar 3.16 Flowdiagram program pairing Bluetooth device aplikasi	31
Gambar 3.17 Flowdiagram disconnect device bluetooth	32
Gambar 3.18 Flowchart pembangkitan PWM.....	33
Gambar 3.19 Contoh segmen program pembangkitan PWM	34
Gambar 3.20 Contoh segmen program auto-fading	35
Gambar 3.21 Flowchart program music synched.....	36
Gambar 3.22 Contoh segmen program music synched	37
Gambar 3.23 Blok diagram sistem pengolahan citra	38
Gambar 3.24 Flowchart Program Pengolahan Citra	39
Gambar 3.25 Segmentasi Program Parsing Data Image Processing	40

Gambar 4.1 Halaman login aplikasi android	41
Gambar 4.2 Halaman menu aplikasi android.....	42
Gambar 4.3 (a) Halaman kontrol led 1 (b) Halaman kontrol led 2 (c) Halaman kontrol led 3	42
Gambar 4.4 (a) Halaman kontrol animasi (b) Halaman kontrol auto- brightness (c) Halaman kontrol auto-fading	43
Gambar 4.5 Antarmuka login aplikasi android	44
Gambar 4.6 Antarmuka pengujian login gagal	44
Gambar 4.7 Antarmuka connect bluetooth	46
Gambar 4.8 Jendela listview device bluetooth.....	46
Gambar 4.9 Antarmuka disconnect bluetooth.....	47
Gambar 4.10 Pengiriman data RGB Raspberry pi	49
Gambar 4.11 Pengambilan data warna dengan Webcam.....	50
Gambar 4.12 Program Parsing Data Arduino	50
Gambar 4.13 Grafik respon animasi LED dengan periode 5 detik	55
Gambar 4.14 Grafik respon animasi LED dengan periode 8 detik	55
Gambar 4.15 Grafik respon animasi LED dengan periode 10 detik	56
Gambar 4.16 Grafik respon LED pada program music-synched	57

DAFTAR TABEL

HALAMAN

Tabel 2.1 AT Command Module Bluetooth HC-05	9
Tabel 3.1 Konfigurasi Port Mikrokontroler	18
Tabel 4.1 Pengujian Antarmuka Menu Area.....	45
Tabel 4.2 Data pengiriman kendali LED	47
Tabel 4.3 Data pengiriman kendali animasi.....	48
Tabel 4.4 Pengujian jarak kontrol dan waktu tanggap bluetooth	51
Tabel 4.5 Pengujian nyala/mati LED RGB	52
Tabel 4.6 Pengujian warna LED RGB	53
Tabel 4.7 Pengujian komunikasi interface android dengan arduino	63
Tabel 4.8 Pengujian Komunikasi Data LED 1 Warna Merah	64
Tabel 4.9 Pengujian Komunikasi Data LED 1 Warna Hijau.....	65
Tabel 4.10 Pengujian Komunikasi Data LED 1 Warna Biru	66
Tabel 4.11 Pengujian Komunikasi Data LED 2 Warna Merah	67
Tabel 4.12 Pengujian Komunikasi Data LED 2 Warna Hijau.....	68
Tabel 4.13 Pengujian Komunikasi Data LED 2 Warna Biru	69
Tabel 4.14 Pengujian Komunikasi Data LED 3 Warna Merah	70
Tabel 4.15 Pengujian Komunikasi Data LED 3 Warna Hijau.....	71
Tabel 4.16 Pengujian Komunikasi Data LED 3 Warna Biru	72
Tabel 4.17 Pengujian komunikasi Raspberry pi dengan Arduino	73
Tabel 4.18 Pengujian program animasi auto-fading periode 5 detik.....	74
Tabel 4.19 Pengujian program animasi auto-fading periode 8 detik.....	75
Tabel 4.20 Pengujian program animasi auto-fading periode 10 detik....	76
Tabel 4.21 Pengujian program music-synched	77

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era globalisasi ini, perkembangan ilmu pengetahuan dan teknologi informasi berkembang sangat pesat sehingga memudahkan dalam melakukan aktifitas. Salah satu contoh pesatnya perkembangan teknologi informasi adalah perkembangan dari *smartphone* yang memberikan dampak besar pada kebiasaan penggunaan perangkat tersebut yang pada awalnya digunakan hanya untuk keperluan komunikasi suara, pesan singkat, pesan elektronik, dan keperluan internet menuju pada penggunaan yang mulai menunjang kebutuhan sehari-hari, baik untuk keperluan bekerja maupun kebutuhan akan hiburan. Tetapi sekarang kemampuan *smartphone* sudah mampu mengotomasi segala sesuatu melalui aplikasinya seperti yang dilakukan oleh komputer. Hal ini menjadikan otomasi lebih mudah dan lebih efisien. Salah satu hal yang dapat diotomasi menggunakan *smartphone* adalah sistem pengaturan warna LED RGB sebagai *Art lighting* dalam industri dibidang desain interior bangunan.

Proses pengendalian akan lebih mudah jika instalasi sistem terintegrasi menjadi satu melalui media *smartphone*. Hal itu dapat memudahkan pengguna dalam mengendalikan pencahayaan sesuai keinginan. Untuk mencapai tujuan tersebut diperlukan sistem kendali secara terpusat dalam perubahan warna cahaya LED RGB agar dihasilkan kombinasi warna sesuai keinginan dan dapat mempertahankan intensitas pencahayaan sesuai *setpoint* yang telah ditentukan. Serta dilengkapi fitur – fitur animasi yang dinamis dan inovatif untuk menambah kelengkapan variasi pengendalian.

Berdasarkan latar belakang tersebut, dirancanglah sebuah sistem pengaturan cahaya LED dengan kombinasi warna RGB. *Art lighting* memerlukan *user interface* untuk membantu pengguna dalam merubah warna cahaya suatu ruang. Salah satu media yang dapat membantu dalam pengendalian tersebut adalah dengan membuat aplikasi yang sudah dirancang di *smartphone* Android menggunakan *MIT App Inventor 2*. *Smartphone* Android dihubungkan dengan mikrokontroler Arduino Mega melalui modul bluetooth HC-05. Aplikasi akan memasukkan kombinasi warna LED RGB melalui sebuah *color picker* yang terpasang pada Android.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan adalah bagaimana membuat aplikasi pengendalian warna LED RGB berbasis android menggunakan App Inventor 2, membuat sistem *image processing* untuk mengakuisisi data RGB serta mengkomunikasikan perangkat mikrokontroler Arduino dengan Raspberry Pi.

1.3 Batasan Masalah

- Pembuatan aplikasi *smartphone* Android menggunakan *App Inventor 2 (AI2)*.
- Pengiriman dan penerimaan data dari *smartphone* Android ke Mikrokontroler Arduino melalui modul *Bluetooth HC-05*.
- Sistem *image processing* menggunakan perangkat Raspberry Pi.
- Pemrosesan data dari *smartphone* Android oleh Mikrokontroler Arduino Mega.
- Menggunakan 3 Buah LED *Stripes* RGB dengan level tegangan 24 Volt.

1.4 Tujuan

Tujuan dari penelitian ini adalah untuk membuat sistem aplikasi pengendalian warna LED RGB secara terpusat berdasarkan data dari *smartphone* Android. Hasil yang diharapkan adalah data yang diterima dari aplikasi Android dapat diproses oleh Mikrokontroler Arduino berupa pembangkitan PWM (*Pulse Width Modulation*) 0 – 255 untuk mengendalikan warna LED.

1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, pemodelan sistem, perancangan algoritma kontrol, simulasi hasil desain, implementasi dan analisis data, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai app inventor, perancangan desain aplikasi, pemrograman *Block-Code App*, karakteristik sensor mikrofon KY-038, komunikasi modul *bluetooth HC-05*, mikrokontroler arduino mega, serta *driver* mosfet. Pada tahap perancangan sistem, pembuatan desain *layout* aplikasi android akan dibentuk dengan platform app inventor 2. Dari pembuatan desain *layout* tersebut, diberikan fungsi instruksi dengan cara memprogram aplikasi dengan kode blok app inventor. Kemudian, aplikasi android di

koneksikan dengan mikrokontroler arduino melalui komunikasi *Bluetooth*. Setelah itu, mikrokontroler Arduino diprogram untuk membangkitkan PWM (*Pulse Width Modulation*) dan beberapa fungsi lainnya. Selanjutnya mengimplementasikan sistem pada *real plant* untuk dilakukan percobaan aplikasi dan pengambilan data. Yang terakhir adalah membuat sistem *image processing* menggunakan perangkat raspberry pi untuk mengenali warna sebuah ojek yang kemudian warna tersebut dibangkitkan pada LED RGB.. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini membahas tentang teori penunjang yang mendukung dalam perencanaan pembuatan alat meliputi pemrograman *App Inventor 2*, teori driver mosfet, aplikasi sensor *sound*, komunikasi *bluetooth*, sistem *image processing* dengan raspberry pi, perancangan mikrokontroler Arduino dan fungsi - fungsi dari prinsip lainnya.

Bab III Perancangan Sistem

Bab ini membahas tentang perencanaan dan perancangan sistem yang meliputi perangkaian driver mosfet, mikrokontroler Arduino, perancangan sensor *sound*, perancangan sistem *image processing* dengan raspberry pi. Serta pembuatan perangkat lunak (*software*) berupa program untuk membangkitkan PWM (*Pulse Width Modulation*) pada Mikrokontroler

Arduino dan program *User Interface* yang dibuat menggunakan *App Inventor 2*.

Bab IV Implementasi dan Analisis Sistem

Bab ini memuat hasil implementasi dan analisis dari hasil tersebut.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan mempermudah pengguna dalam mengendalikan warna LED RGB secara terpusat menggunakan *smartphone* Android.

BAB II

TEORI DASAR

2.1 App Inventor

App Inventor adalah sebuah tool untuk membuat aplikasi android yang berbasis *visual block programming*, sehingga pengguna bisa membuat aplikasi tanpa melakukan *coding*. *Visual block programming* maksudnya adalah dalam penggunaannya user akan melihat, menggunakan, menyusun dan *drag-drops* “blok” yang merupakan simbol-simbol perintah dan fungsi *event handler* tertentu dalam membuat aplikasi, dan secara sederhana bisa disebut tanpa menuliskan kode program. Aplikasi *App Inventor* ini pada dasarnya adalah aplikasi yang disediakan oleh *google* dan sekarang di-maintenance oleh *Massachusetts Institute of Technology* (MIT). Aplikasi ini selesai dibuat pada 12 juli 2010 dan dirilis untuk public pada 31 Desember 2011. *App Inventor* sekarang dipegang oleh *MIT Centre for Mobile Learning* dengan nama *MIT App Inventor*.



Gambar 2.1 Tampilan Halaman Desain Aplikasi App Inventor

Pada lingkungan kerja App Inventor ini terdapat beberapa komponen yang terdiri dari :

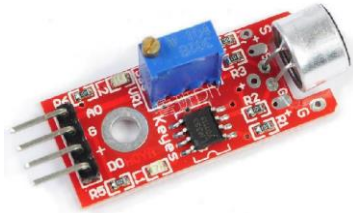
1. Komponen Desainer

Komponen desainer berjalan pada browser yang digunakan untuk memilih komponen yang dibutuhkan dan mengatur property-nya. Pada komponen desainer sendiri terdapat 5 bagian, yaitu palette, viewer, component, media dan properties.

- a. Palette : list komponen yang bisa digunakan.
 - b. Viewer : untuk menempatkan komponen dan mengaturnya sesuai tampilan yang diinginkan.
 - c. Component : tempat list komponen yang dipakai pada project.
 - d. Media : mengambil media audio atau gambar untuk project.
 - e. Properties : mengatur properties komponen yang digunakan, seperti width, height, name, dll.
2. Blok Editor
Blok editor berjalan di luar browser dan digunakan untuk membuat dan mengatur behaviour dari komponen-komponen yang dipilih dari komponen desainer.
 3. Emulator
Emulator digunakan untuk menjalankan dan mencoba project yang dibuat. Jadi sebelum mengunduh aplikasi tersebut untuk diinstal di smartphone, pengguna dapat mencoba hasil dari aplikasi tersebut terlebih dahulu.

2.2 Sensor Suara KY-038

Sensor suara adalah sensor yang cara kerjanya merubah besaran suara menjadi besaran listrik, dan dipasaran sudah begitu luas penggunaannya.. Sensor ini digunakan untuk mendeteksi suara di area sekitar. Intensitasnya bisa di atur manual dengan cara mengubah nilai resistor pada sensor. Komponen yang termasuk dalam Sensor suara yaitu *electric condenser microphone* atau mic kondenser. Prinsip kerja ECM adalah getaran suara yang diterima oleh dielektrik berupa membran tipis di dalam ECM akan menyebabkan perubahan nilai kapasitasnya. *Condenser mic* bekerja berdasarkan diafragma atau susunan *backplate* yang harus tercatu oleh listrik membentuk sound-sensitive capacitor. Gelombang suara yang masuk ke *microphone* akan menggetarkan komponen diafragma ini. Letak dari diafragma ditempatkan di depan sebuah *backplate*. Susunan dari elemen ini membentuk sebuah kapasitor yang biasa disebut juga kondenser. Variasi akan lebar *space* antara diafragma dengan *backplate* terjadi dikarenakan adanya pergerakan diafragma relatif terhadap backplate yang disebabkan oleh adanya tekanan suara yang mengenai diafragma. Hal ini akan menghasilkan sinyal elektrik dari gelombang suara yang masuk ke *condenser microphone*.



Gambar 2.2 Sensor Mikrofon KY-038

Berdasarkan gambar 2.2, konfigurasi pin sensor mikrofon adalah sebagai berikut :

1. Pin 5 V : Dicapai daya sebesar 5 Volt.
2. Pin GND : Pin *Ground*
3. DO : Pin hasil keluaran sinyal sensor yang berbentuk sinyal digital.
4. AO : Pin keluaran sinyal analog

2.3 Modul Bluetooth HC-05

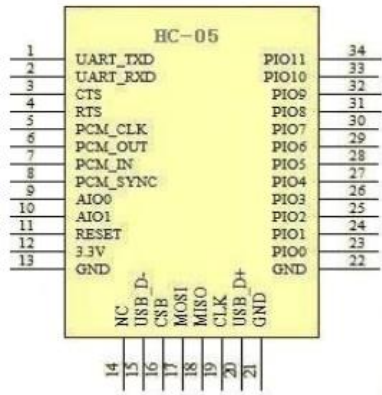
Bluetooth adalah protokol komunikasi wireless yang bekerja pada frekuensi radio 2.4 GHz untuk pertukaran data pada perangkat bergerak seperti PDA, laptop, HP, dan lain-lain1 . Salah satu hasil contoh modul Bluetooth yang paling banyak digunakan adalah tipe HC-05. modul Bluetooth HC-05 merupakan salah satu modul Bluetooth yang dapat ditemukan dipasaran dengan harga yang relatif murah. Modul Bluetooth HC-05 terdiri dari 6 pin konektor, yang setiap pin konektor memiliki fungsi yang berbeda - beda. Untuk gambar module bluetooth dapat dilihat pada gambar 2.3 dibawah ini:



Gambar 2.3 Modul *Bluetooth* HC-05

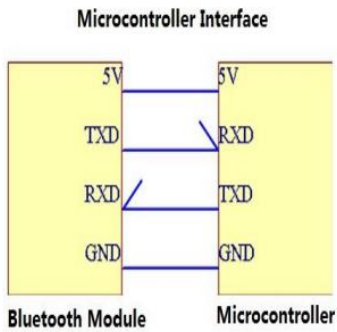
Modul Bluetooth HC-05 dengan supply tegangan sebesar 3,3 V ke pin 12 modul Bluetooth sebagai VCC. Pin 1 pada modul Bluetooth

sebagai transmitter. kemudian pin 2 pada Bluetooth sebagai receiver. Berikut merupakan konfigurasi pin bluetoooth HC-05 ditunjukkan pada gambar 2.4 dibawah ini:



Gambar 2.4 Konfigurasi Pin HC-05

Berikut merupakan Bluetooth-to-Serial-Module HC-05 dapat dilihat pada gambar 2.5 dibawah ini:



Gambar 2.5 Bluetooth-to-Serial-Module

Bluetooth HC-05 merupakan module Bluetooth yang bisa menjadi slave ataupun master hal ini dibuktikan dengan bisa memberikan notifikasi untuk melakukan pairing keperangkat lain, maupun perangkat lain tersebut yang melakukan pairing ke module Bluetooth CH-05. Untuk mengeset perangkat Bluetooth dibutuhkan perintah-perintah AT Command yang mana perintah AT Command tersebut akan di respon oleh perangkat Bluetooth jika modul Bluetooth

tidak dalam keadaan terkoneksi dengan perangkat lain. Tabel 2.1 dibawah adalah tabel AT Command Module Bluetooth CH-05. Keterangan AT Command Module Bluetooth CH-05 dapat dilihat pada tabel 2.1 berikut:

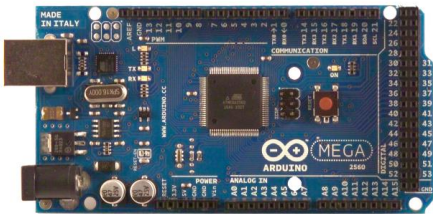
No	Perintah	Kirim	Terima	Keterangan
1.	Test Komunikasi	AT	ON	-
2.	Ganti Nama Bluetooth	AT+NAMEnamaBT	OKnamaBT	-
3.	Ubah Pin Code	AT+PINxxxx	OKsetpin	Xxxx digit key
4.	Ubah Baudrate	AT+BAUD1 AT+BAUD2 AT+BAUD3 AT+BAUD4 AT+BAUD5 AT+BAUD6	OK1200 OK2400 OK4800 OK9600 OK19200 OK38400	1——1200 2——2400 3——4800 4——9600 5——19200 6——38400 7——57600 8——115200

Tabel 2.1 AT Command Module Bluetooth HC-05

2.4 Mikrokontroler Arduino Mega

Mikrokontroler adalah sebuah komputer kecil disuatu sirkuit terpadu yang berisi tentang inti prosesor, memori dan *input/output* yang telah diprogram. Program disimpan dalam bentuk RAM, Nor Flash, OTP ROM yang disertakan dalam chip.

Salah satu jenis mikrokontroler dari tipe Atmel AVR adalah ATmega 2560. ATmega2560 adalah mikrokontroller keluaran dari atmel yang mempunyai arsitektur RISC (*Reduce Instruction Set Computer*) yang setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (*Completed Instruction Set Computer*).



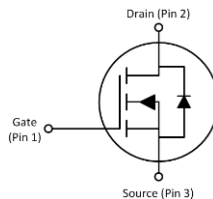
Gambar 2.6 Mikrokontroler ATmega 2560

Secara garis besar mikrokontroller ini memiliki arsitektur sebagai berikut:

1. 135 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus clock.
2. 32 x 8-bit register.
3. Kecepatan mencapai 16 MIPS dengan clock 16 MHz.
4. 256 KB *Flash memory* dan 8 KB pada bootloader.
5. Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 4 KB sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
6. Memiliki SRAM (Static Random Access Memory) sebesar 8 KB.
7. Memiliki pin I/O digital sebanyak 54 pin 14 diantaranya PWM (*Pulse Width Modulation*) output.

2.5 Transistor Mosfet

MOSFET (*Metal Oxide Semiconductor Field Effect Transistor*) merupakan salah satu jenis transistor yang memiliki impedansi masukan (gate) sangat tinggi (Hampir tak berhingga) sehingga dengan menggunakan MOSFET sebagai saklar elektronik, memungkinkan untuk menghubungkannya dengan semua jenis gerbang logika. Dengan menjadikan MOSFET sebagai saklar, maka dapat digunakan untuk mengendalikan beban dengan arus yang tinggi dan biaya yang lebih murah daripada menggunakan transistor bipolar. Untuk membuat MOSFET sebagai saklar maka hanya menggunakan MOSFET pada kondisi saturasi (ON) dan kondisi *cut-off* (OFF).



Gambar 2.7 Simbol MOSFET Channel N

2.6 Light Emitting Diode RGB

LED peralatan menjadi sumber penerangan utama di berbagai bidang masyarakat. Proyektor portable dan senter portable, lampu dan monitor layar pencahayaan alas yang dibuat menggunakan LED. Popularitas seperti elemen semikonduktor ini tersedia mereka

kehandalan , ketahanan dan efisiensi. Struktur LED membedakan 2 jenis strip LED:

- a. Monokrom
LED yang menghasilkan satu jenis warna ketika diberi *supply*
- b. RGB LED
LED yang terdiri dari penuh warna *light-emitting diode* mampu memberikan penerangan dari sejumlah besar warna dan nuansa.



Gambar 2.8 RGB LED Strip

Untuk menghubungkan beberapa strip RGB, dianjurkan untuk menggunakan amplifier. Untuk penguat akan membutuhkan sumber tegangan suplai yang sesuai. Hal ini dimungkinkan untuk melaksanakan pasokan listrik dari adaptor sudah terinstal jika diperebutkan kekuatannya, atau untuk menghubungkan power supply tambahan.

Wiring LED strip akan menjadi berikutnya:

Untuk melakukan koneksi yang tepat untuk tape controller, Anda dapat menggunakan warna dan literal menandai. Standar kabel pita keluar berwarna, masing-masing terhubung ke konektor khusus:

- a. Red - surat R.
- b. Hijau - surat G.
- c. Blue - surat B.
- d. Putih - V +.

2.7 Raspberry Pi 3

Raspberry Pi merupakan *single board computer* (SBC) seukuran dengan sebuah kartu kredit dengan berbagai fungsi yang dapat dilakukannya. Raspberry Pi menggunakan sistem operasi Raspbian. Raspberry pi 3 model B memiliki *processor* dengan spesifikasi *Arm8 64-bit Quad Core Processor powered Single Board Computer running at 1.2 GHz*. Raspberry pi model B menggunakan *micro SDCard* sebagai media penyimpanannya. Selain itu Raspberry juga dilengkapi 4 buah

port USB untuk tipe B, konektor HDMI, Raspberry Pi dilengkapi dengan port *ethernet*. Pada Raspberry Pi tidak disediakan *switch power*. Port *micro USB* pada Raspberry Pi digunakan sebagai *power supply*, penggunaan *micro USB* dikarenakan murah dan mudah didapatkan. Raspberry pi 3 model B membutuhkan *supply* sebesar 5 volt dengan arus minimal 700mA.

Pemrograman yang digunakan pada raspberry pi untuk melakukan perintah menggunakan Bahasa python. Python merupakan Bahasa pemrograman yang *freeware* atau perangkat ebas dalam arti sebenarnya, tidak ada Batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan codenya, *debugger* dan *profiler*, antarmuka yag terkandung didalamnya untuk pelayanan antarmuka, fungsi sistem, GUI, dan basis datanya.



Gambar 2.9 Raspberry Pi 3 Model B

Pada raspberry pi disediakan pin-pin input/output (IO), diantaranya adalah :

1. *General Purpose Input Output (GPIO)*

Pin-pin tersebut dapat digunakan untuk membaca input dari tombol serta *switches* untuk mengontrol *actuator* seperti LED, Relay dan motor yang difungsikan sebagai input dan output data digital. GPIO merupakan sederet pin yang terdiri dari 26 pin dengan berbagai fungsi diantaranya:

3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Gambar 2.10 Pin GPIO Raspberry Pi 3

2. Display Serial Interface (DSI) Connector

Konektor ini dapat digunakan dengan menggunakan kabel pita tipis 15 pin sebagai penghubung dengan LCD Display.

3. Camera Serial Interface (CSI) Connector

Port ini berfungsi sebagai penghubung langsung antara raspberry pi dengan sebuah modul camera.

Pada 8 Maret 2012 Yayasan Pi Raspberry merilis Raspberry Pi Fedora Remix direkomendasikan sebagai distribusi Linux, yang dikembangkan di Seneca College di Kanada. Yayasan ini berniat untuk membuat situs Web App Store bagi orang untuk program pertukaran. Slackware ARM (secara resmi ARMEdslack) versi 13.37 dan kemudian berjalan pada Raspberry Pi tanpa modifikasi. 128–496 MB dari memori yang tersedia di Raspberry Pi adalah dua kali minimum 64 MB yang diperlukan untuk menjalankan Slackware Linux pada sistem ARM atau i386. (Sementara Slackware dapat memuat dan menjalankan GUI, yang dirancang untuk dijalankan dari shell).

Fluxbox window manager berjalan di bawah X Window System memerlukan tambahan 48 MB RAM. Selain itu, pekerjaan yang sedang dilakukan pada distribusi Linux seperti IPFire, OpenELEC, Raspbmc dan XBMC membuka sumber digital media center. Eben Upton secara terbuka mendekati RISC OS pada bulan Juli 2011 untuk menanyakan tentang bantuan dengan port potensial. Adrian Lees di Broadcom sejak

itu bekerja pada port, dengan karyanya yang disebutkan dalam sebuah diskusi tentang driver grafis.

Pada 24 Oktober 2012 Yayasan Raspberry Pi mengumumkan bahwa "semua kode driver VideoCore yang berjalan pada ARM" telah dirilis sebagai perangkat lunak bebas di bawah lisensi BSD-style, membuat "multi media pertama berbasis ARM multimedia SoC dengan banyak-fungsional, vendor menyediakan (sebagai lawan dari parsial, reverse rekayasa) sepenuhnya opensource driver", meskipun klaim ini tidak diterima secara universal.

2.8 Open CV

Open CV (Open Computer Vision) adalah sebuah API (Application Programming Interface) Library yang sudah sangat familiar pada Pengolahan Citra Computer Vision.

Computer Vision itu sendiri adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia. Dengan vision tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari Computer Vision adalah Face Recognition, Face Detection, Face/Object Tracking, Road Tracking, dll.

OpenCV adalah sebuah API yang dikembangkan oleh perusahaan INTEL. Device yang telah menggunakan API ini salah satunya KINECT XBOX. Namun sayangnya XCode tidak menyediakan OpenCV Dynamic Framework.

OpenCV adalah library Open Source untuk Computer Vision untuk C/C++, OpenCV didesain untuk aplikasi real-time, memiliki fungsi-fungsi akuisisi yang baik untuk image/video. OpenCV juga menyediakan interface ke Integrated Performance Primitives (IPP) Intel sehingga jika anda bisa mengoptimasi aplikasi Vision anda jika menggunakan prosesor Intel.

OpenCV sendiri terdiri dari 5 library, yaitu :

1. CV : untuk algoritma Image processing dan Vision.
2. ML : untuk machine learning library
3. Highgui : untuk GUI, Image dan Video I/O.
4. CXCORE : untuk struktur data, support XML dan fungsi grafis.
5. CvAux

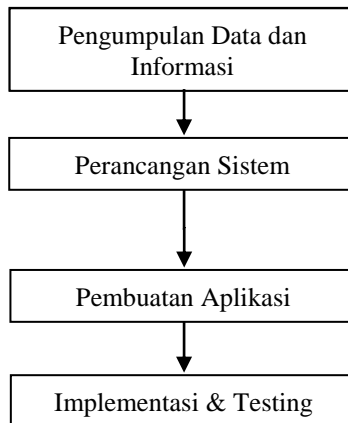
BAB III

PERANCANGAN SISTEM

Pada bab ini dibahas mengenai perancangan perangkat lunak (*software*). Hal tersebut guna mewujudkan tugas akhir yang berjudul “Rancang Bangun Aplikasi Sistem Kendali Warna Led RGB Terpusat Berbasis Android”.

3.1 Kerangka Konseptual

Kerangka konseptual adalah suatu bentuk cara berpikir yang dapat digunakan sebagai pendekatan dalam memecahkan suatu masalah. Biasanya kerangka pada suatu penelitian menggunakan pendekatan ilmiah dan memperlihatkan hubungan antar variabel dalam proses analisisnya.



Gambar 3.1 Kerangka Konseptual

1. Pengumpulan Data dan Informasi

Penelitian ini diawali dengan pengumpulan data dan informasi mengenai pengendalian cahaya, pembuatan desain antar muka pada APP Inventor, pemrograman blok diagram, pemrograman Arduino, serta media komunikasi yang bias dikombinasikan dengan Android.

2. Perancangan Sistem

Perancangan sistem merupakan rancang bangun sistem yang dibuat terdiri dari komponen perangkat lunak, driver mosfet, mikrokontroler, dan perangkat komunikasi. Sistem ini dibuat berdasarkan informasi yang telah didapat melalui research tentang teknologi mobile komunikasi Android, *Bluetooth*, APP Inventor dan Arduino.

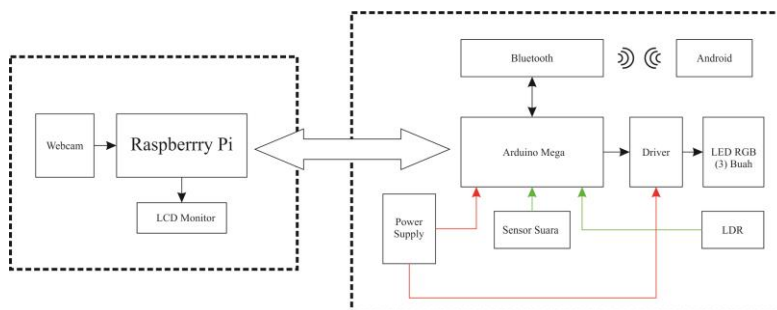
3. Pembuatan Aplikasi

Aplikasi yang akan dibuat menggunakan APP Inventor sebagai interface pengendalian warna LED berbasis Android.

4. Implementasi dan Testing

Aplikasi yang telah dibuat kemudian akan diuji dan diimplementasikan kepada sebuah ruangan yang memiliki 3 buah LED RGB, perangkat Android yang sudah terinstall software dan sensor cahaya untuk mengambil data R, G, B, yang linier dengan tegangan keluaran driver mosfet.

3.2 Blok Fungsional Sistem



Gambar 3.2 Blok Diagram Fungsional Sistem

Sistem pengendali LED ini menggunakan smartphone untuk mengatur perubahan warna pada LED. Tegangan yang dibutuhkan LED RGB berkisar 0-24 Volt, maka diperlukan rangkaian driver untuk merubah variable tegangan 0 – 24 Volt. Pusat pengendali LED RGB menggunakan Mikrokontroler. Sumber tegangan Mikrokontroler berasal

dari power supply yang diregulasi dengan dc to dc voltage regulator sebesar 5 Volt. Power supply ini berperan penting dalam pendistribusian daya ke bagian system yang membutuhkan listrik agar dapat bekerja. Mikrokontroler menghasilkan PWM untuk mengatur switch mosfet pada rangkaian driver. Eksekusi perintah untuk menghidupkan output – output yang ada dilakukan pada aplikasi yang terdapat pada Android, aplikasi akan mengkonversi interface yang dipahami oleh menggunakan kedalam bentuk program untuk perintah I/O terhadap pin pada arduino, sebelum akhirnya perintah akan dikirim melalui Bluetooth ke Arduino. Arduino melakukan komunikasi serial USB dengan perangkat Raspberry Pi 3 dimana perangkat raspberry terhubung dengan *webcam* sebagai masukan dari program pengolahan citra (*image processing*).

Mikrokontoller Arduino Mega menjadi otak atau salah satu inti dalam system ini, Arduino di program menggunakan IDE nya sendiri, yang berbasis Open Source dan menggunakan bahasa pemrograman C++, program perintah yang telah di-*compile* kemudian di upload ke dalam Arduino, yang kemudian menjadikan Arduino sebagai otak yang siap untuk menghubungkan Input dan Output dari Sistem ini sebagai penerima perintah, eksekutor, dan sebagai pengirim *feedback* ke android device. Pada sistem ini terdiri dari 3 buah LED RGB yang akan menerima perintah dari interface android. LDR (*Light dependent resistor*) akan melakukan pengukuran intensitas cahaya ruangan sebagai *feedback* dari sistem control loop tertutup untuk menghasilkan pengendalian intensitas cahaya LED otomatis dengan metode PID. Sistem akan interaktif terhadap pencahayaan ruangan sekitar sehingga dapat mengatur perubahan cahaya sesuai dengan *setpoint* yang diberikan.

3.3 Konsep Perancangan Perangkat Lunak

Pada tugas akhir ini penulis membuat sistem pengendali warna LED yang dikontrol melalui smartphone berbasis android. Agar smartphone dapat memberi perintah/masukkan ke mikrokontroler arduino, harus dibuat aplikasi yang dapat menjalankan perintah berupa angka/huruf yang di transfer melalui jalur bluetooth. Dalam pembuatan aplikasi berbasis android ini, penulis menggunakan App Inventor. App Inventor merupakan sebuah aplikasi web *open-source* asli yang disediakan oleh Google dan sekarang dikelola oleh Massachusetts Institute of Technology (MIT). Dalam pembuat aplikasi android dengan App Inventor terdapat tiga langkah yaitu :

1. Menginstal aplikasi Java pada komputer.
2. Membuat desain aplikasi android dengan komponen desainer yang ada pada App Inventor.
3. Membuat perintah pada *block editor* dengan mengatur komponen-komponen yang kita pilih dari komponen desainer.

Setelah aplikasi telah dibuat, perintah – perintah yang dikirim dari aplikasi tersebut akan diproses oleh mikrokontroler Arduino. Maka diperlukan pemrograman mikrokontroler Arduino untuk membuat perintah eksekusi pengendalian warna LED RGB. Untuk memrogram Arduino, dibutuhkan software Arduino IDE sebagai media penulisan program Arduino berbahasa C/C++. Kemudian program tersebut diupload pada mikrokontroler Arduino.

3.4 Konfigurasi Pin Mikrokontroller

Mikrokontroler digunakan sebagai pusat kendali warna LED RGB. Mikrokontroler yang digunakan adalah Arduino Mega. Arduino Mega adalah sebuah board mikrokontroler didasarkan pada ATmega 2560. Pada sistem pengaturan warna LED RGB digunakan beberapa pin mikrokontroler dengan rancangan sesuai pada Tabel 3.1.

Tabel 3.1 Konfigurasi *Port* Mikrokontroler

No	Pin Arduino	Keterangan
1	Pin 14	Komunikasi TX
2	Pin 15	Komunikasi RX
3	Pin 2	PWM Red 1
4	Pin 3	PWM Red 2
5	Pin 5	PWM Red 3
6	Pin 6	PWM Green 1
7	Pin 7	PWM Green 2
8	Pin 8	PWM Green 3
9	Pin 9	PWM Blue 1
10	Pin 10	PWM Blue 2

11	Pin 11	PWM Blue 3
12	Pin 50	Digital Input Sensor Suara
13	Pin A8	Analog Input Sensor Suara
14	Pin A0	Analog Input LDR
15	VCC, GND	Supply Bluetooth & Sensor Mikrofon
16	Serial USB	Komunikasi dengan Raspberry Pi

3.5 Perancangan Aplikasi Android dengan APP Inventor

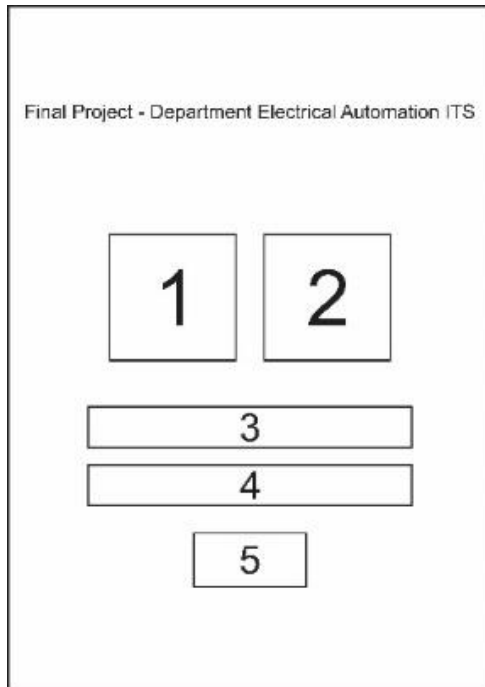
Dalam pembuatan sistem ini, di gunakan ponsel atau smartphone dengan sistem operasi android untuk mengontrol warna LED RGB sebagai *Art Lighting*. Perangkat dengan sistem operasi android ini mengirim perintah untuk mengontrol kombinasi warna LED RGB dalam ruangan, dengan bluetooth (yang sudah terhubung dengan perangkat dan program arduino) memancarkan sinyal untuk komunikasi data dan di pair oleh aplikasi. Uji coba dari perancangan dan pembuatan aplikasi android ini digunakan ponsel/smartphone dengan sistem operasi android , dalam penelitian ini menggunakan semua type smartphone.

App Inventor sebagai platform pembuatan software android berbasis *visual block programming* sangat mudah dan efektif digunakan bagi user. Langkah pertama untuk mengakses platform ini adalah dengan membuat akun *Google* terlebih dahulu dan login kedalamnya. Setelah akun dibuat kemudian masuk ke browser dan ketik appinventor.mit.edu untuk memulai App Inventor.

Perancangan desain aplikasi terdiri dari beberapa grup layer, layer pertama adalah login screen, kemudia layer kedua adalah menu screen yang kemudian terdapat grup layer didalamnya yakni 3 led screen dan animation screen.

1. Perancangan Desain Login Screen

Desain perangkat lunak android pada app inventor diawali dengan halaman login terlebih dahulu. Hal ini bertujuan untuk menjaga *operator-privacy* agar tidak diakses oleh pengguna umum. Berikut sketsa halaman login pada sistem kendali LED RGB.



Gambar 3.3 Sketsa halaman login *software* android

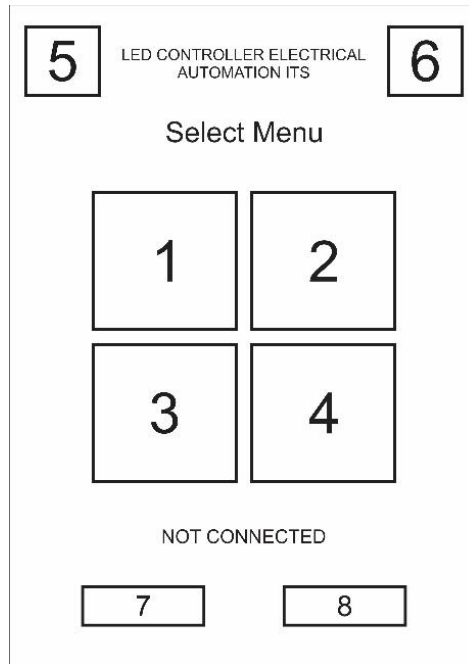
Berikut keterangan gambar 3.3 :

1. Layout 1, Logo ITS
2. Layout 2, Logo Departemen Elektro Otomasi ITS
3. TextBox1, Username
4. TextBox2, Password
5. Button1, Submit/Login

Sketsa pada Gambar 3.3 merupakan gambar dari halaman depan aplikasi android yang dibuat dengan App Inventor. Untuk mengakses halaman login, pengguna diminta untuk memasukkan kode *username* dan *password*. Ketika kode tersebut benar, maka aplikasi akan beralih pada menu selanjutnya. Jika kode salah maka akan muncul *alert-message* sebagai tanda kode *username* dan *password* salah.

2. Perancangan Desain Menu Screen

Halaman aplikasi yang kedua adalah menu screen dimana menu ini berfungsi untuk memilih nomor LED RGB yang hendak dikendalikan dan memilih fitur animasi *auto fading* pada pengaturan warna LED. Berikut sketsa halaman menu screen pada aplikasi android :



Gambar 3.4 Sketsa halaman menu *software* android

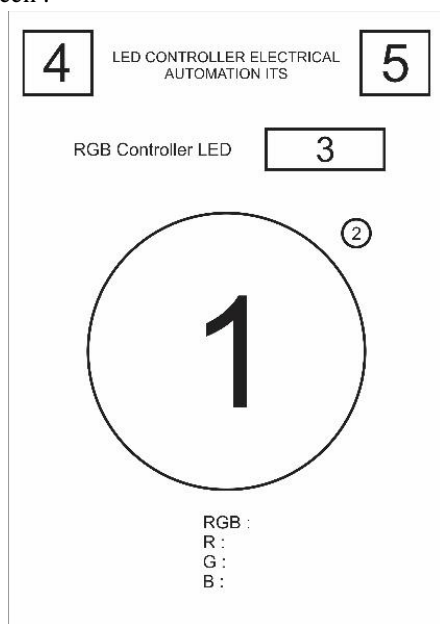
Berikut keterangan gambar 3.4 :

1. Button 2, Led Screen 1
2. Button 3, Led Screen 2
3. Button 4, Led Screen 3
4. Button 5, Animaton Screen
5. Layout 3, Logo ITS
6. Layout 4, Logo Departemen Elektro Otomasi ITS
7. Button 6, Connecting Bluetooth
8. Button 7, Disconnecting Bluetooth

Tampilan menu *screen* dibuat untuk memilih nomor kendali pada LED RGB. Terdapat 4 Fitur untuk melakukan pengaturan LED RGB , yakni Led Screen1, Led Screen2, Led Screen3 dan Animation Screen. Pengguna dapat mengakses 4 macam fitur yang dapat dipilih pada menu.

3. Perancangan Desain Led Screen

Kendali warna LED RGB dapat diaplikasikan pada halaman led screen dimana terdapat 2 pengaturan yakni menghidupkan atau mematikan LED dan mengubah warna LED. Berikut sketsa desain halaman led screen :



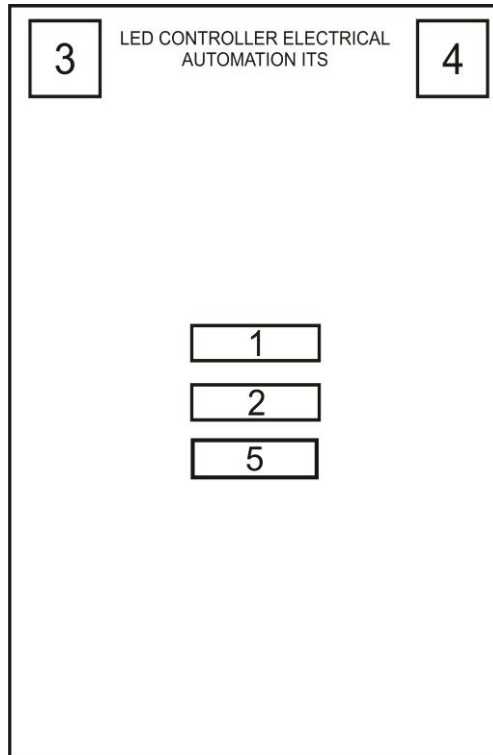
Gambar 3.5 Sketsa halaman led *software* android

Berikut keterangan gambar 3.5 :

1. Canvas 1, Wheel (Color Picker)
2. Ball 1, Color Status
3. Button 8, LED Switch
4. Layout 5, Logo ITS
5. Layout 6, Logo Departemen Elektro Otomasi ITS

4. Perancangan Desain Animation Screen

Perangkat lunak android kendali LED RGB dilengkapi dengan beberapa fitur diantaranya adalah fitur animasi dan sinkronisasi dengan music . Berikut sketsa desain halaman animasi :



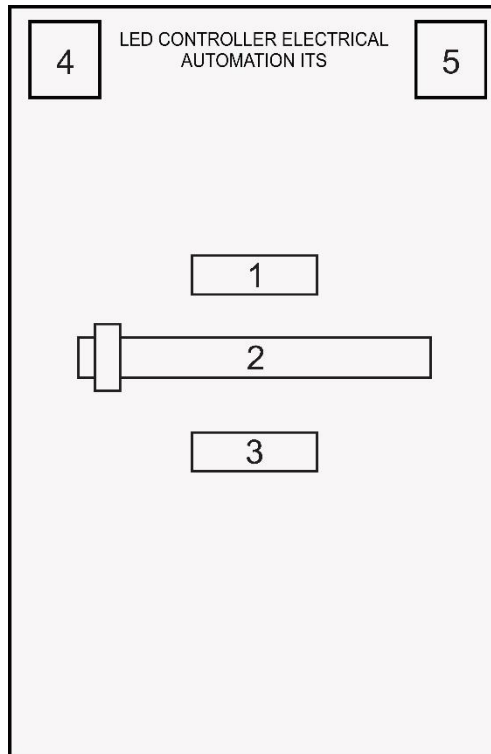
Gambar 3.6 Sketsa halaman animasi *software* android

Berikut keterangan gambar 3.6 :

1. Button 9, Automatic Mode Switch
2. Button 10, Music Synched Mode Switch
3. Layout 7, Logo ITS
4. Layout 8, Logo Departemen Elektro Otomasi ITS
5. Button 11, Animation Mode Switch

5. Perancangan Desain Automatic Screen

Perangkat lunak android kendali LED RGB dapat mengendalikan intensitas cahaya ruangan otomatis sesuai dengan perubahan cahaya. Berikut sketsa halaman desain mode otomatis :



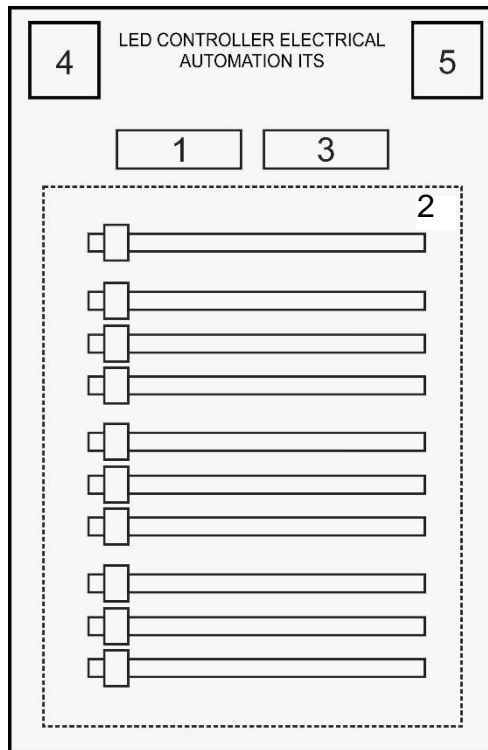
Gambar 3.7 Sketsa halaman mode otomatis android

Berikut keterangan gambar 3.7 :

1. Button 12, Enable/ Disable Mode Pengaturan Cahaya Otomatis
2. Slider 1, Setpoint
3. Button 13, Button Set Nilai Setpoint
4. Layout 9, Logo ITS
5. Layout 10, Logo Departemen Elektro Otomasi ITS

6. Perancangan Desain Auto Fading Screen

Perangkat lunak android kendali LED RGB dilengkapi dengan animasi *auto fading*. LED akan berganti warna dengan rentang waktu tertentu. Berikut sketsa desain halaman animasi untuk mengaktifkan *auto fading* :



Gambar 3.8 Sketsa halaman mode *auto-fading* android

Berikut keterangan gambar 3.8 :

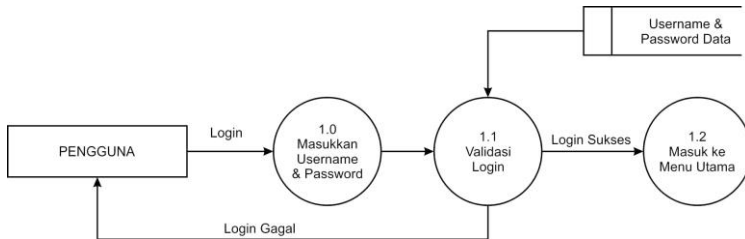
1. Button 14, Enable/ Disable Mode Animasi Auto-fading
2. Slider 2 - 10, Parameter Animasi
3. Button 15, Set Nilai Parameter
4. Layout 11, Logo ITS
5. Layout 12, Logo Departemen Elektro Otomasi ITS

3.6 Pemrograman Kode Blok Aplikasi

Setelah membuat desain aplikasi, berikutnya adalah mengatur fungsi dari tiap komponen dengan membuka *Block Editor*. Dengan *block editor* ini aplikasi ditujukan agar dapat memberikan fungsi pada setiap komponen interface.

1. Login Program

Kode blok dibuat bertujuan untuk mengkonfirmasi ketepatan kode *username* dan *password* sehingga dapat mengakses halaman login dan mengalihkannya ke menu pengoperasian kendali warna LED RGB. Berikut *source-code* login program pada App Inventor :



Gambar 3.9 Flowdiagram halaman login aplikasi android

Berikut penjelasan *Flowdiagram* pada Gambar 3.9 :

1. Pengguna masuk kedalam halaman login. Pengguna memasukkan username dan password pada *textbox*.
2. Ketika button Login ditekan, maka program akan memeriksa kebenaran kode *username* dan *password* yang telah tersimpan.
3. Jika username “admin” dan password “akusayangkamuu” program kemudian membuka screen yang bernama “Menu”. Jika *username* dan *password* tidak benar, *alert-message* akan muncul dengan teks “Invalid Password”.

2. Menu Program

Tampilan menu di program untuk diberikan perintah agar *user* dapat memilih halaman lain yang terdapat dalam menu screen. Pada menu screen terdapat 4 pilihan menu yakni led1, led2, led3, adan animation. Berikut *source-code* program menu pada App Inventor :

```

when Menu.Initialize
do
  set ActivityStarter1.Action to "android.bluetooth.adapter.action.REQUEST_ENABLE"
  call ActivityStarter1.StartActivity
  set led1.Enabled to false
  set led2.Enabled to false
  set led3.Enabled to false
  set animation.Enabled to false
  set Disconnect.Enabled to false

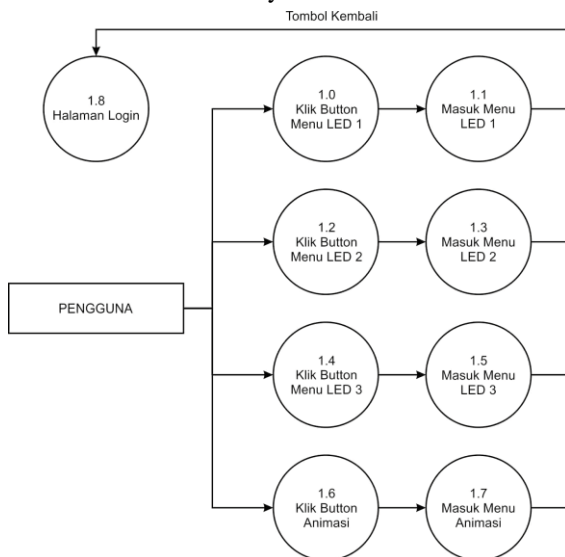
```

Gambar 3.10 Source-Code halaman menu aplikasi android

Berikut penjelasan kode blok pada Gambar 3.10 :

1. Ketika pengguna masuk kedalam halaman menu, sistem akan *start* aplikasi untuk mengaktifkan *request bluetooth* pada perangkat android.
2. Jika *bluetooth* tidak aktif pada perangkat android, maka menu juga tidak aktif. Sehingga pengguna tidak dapat mengakses menu.

Dalam pemilihan menu, *button* diberikan fungsi untuk berpindah dari menu satu ke menu lainnya dengan cara mengatur visibilitas *layout* button. Berikut *source-code* nya :



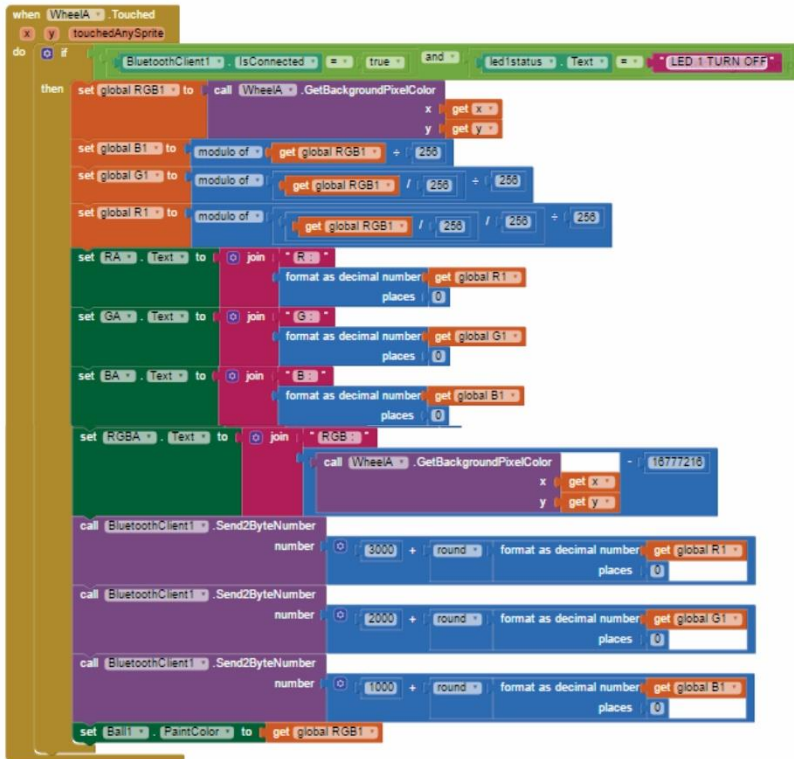
Gambar 3.11 Flowdiagram pemilihan menu aplikasi android

Berikut penjelasan *flowdiagram* pada Gambar 3.11 :

1. Ketika button LED di tekan, maka pengguna masuk pada halaman kendali LED. Halaman LED berjumlah 3. Jika button animasi ditekan, maka pengguna masuk pada halaman animasi.
2. Ketika pengguna menekan tombol *back*, maka pengguna kembali kepada halaman login.

3. Color Picker Program

Program color picker termasuk dalam halaman menu led. Tujuan program ini adalah mengambil nilai RGB pada sebuah palet warna yang kemudian nilai tersebut dikirim pada perangkat mikrokontroler arduino. Berikut *source-code* program color picker :



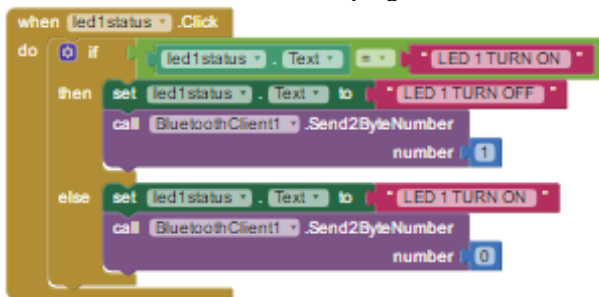
Gambar 3.12 Source-Code program color picker aplikasi android

Berikut penjelasan kode blok pada Gambar 3.12 :

1. Palet warna yang akan diambil nilai RGB nya dinyatakan dalam bentuk canvas. Bentuk canvas memungkinkan sistem dapat responsif terhadap pengguna.
2. Sebelum program mengeksekusi sebuah perintah, dilakukan pengecekan apakah *Bluetooth* terkoneksi dengan android dan status LED RGB adalah ON (menyala).
3. Jika memenuhi, kemudian program akan mengambil data palet warna berupa data koordinat x dan y dan akan disimpan pada variabel RGB.
4. Data koordinat tersebut dikonversikan kedalam bentuk nilai R, G, dan B yang masing – masing nilai disimpan pada variable R1, G1, dan B1.
5. Variabel R1, G1, B1 dipanggil kembali untuk mengubah teks dan ditampilkan nilainya pada aplikasi android.
6. Data nilai RGB kemudian dikirim kepada perangkat mikrokontroler Arduino melalui komunikasi *Bluetooth*. Data yang dikirim dalam bentuk *byte*.
7. Nilai RGB yang memiliki range 0-255 dikirim dengan penambahan *byte* yang berbeda setiap R, G, dan B. Hal ini bertujuan untuk membedakan antara nilai R, G, dan B pada perangkat mikrokontroler agar dapat mengeksekusi nilai tersebut.
8. Nilai R + 3000, nilai G + 2000, dan nilai B + 1000 dikirim dengan format desimal.

4. LED Switch Program

Program LED switch bertujuan untuk menyalakan atau mematikan LED RGB. Berikut *source-code* program LED Switch :



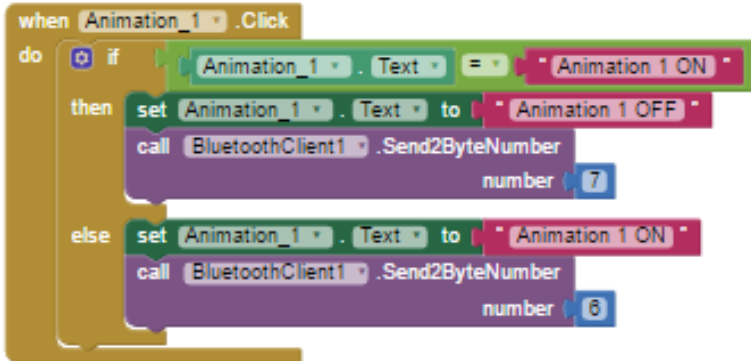
Gambar 3.13 *Source-Code* program LED Switch aplikasi android

Berikut penjelasan kode blok pada Gambar 3.13 :

1. Jika LED ON, program akan mengirim data *byte* 1 pada perangkat mikrokontroler Arduino.
2. Jika LED OFF, program akan mengirim data *byte* 0 pada perangkat mikrokontroler Arduino.

5. Animation Program

Program animasi bertujuan untuk mengaktifkan fitur animasi pada LED RGB. Warna LED RGB akan memiliki pola pergantian warna dari warna merah ke hijau, warna hijau ke biru, kemudian kembali lagi ke warna merah. Perubahan warna tersebut akan berjalan terus menerus hingga program animasi dinonaktifkan. Berikut *source-code* program untuk mengaktifkan animasi :



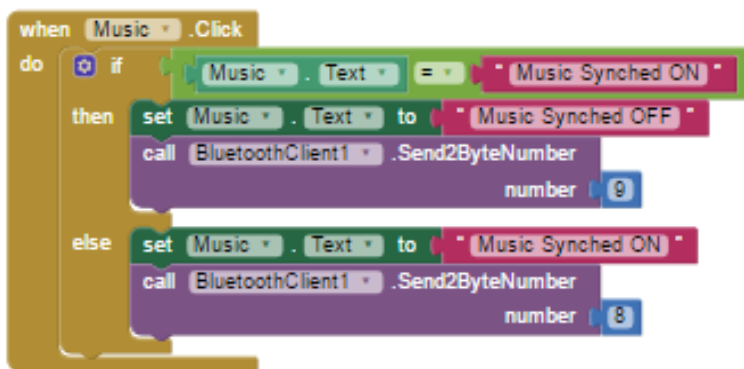
Gambar 3.14 *Source-Code* program animasi aplikasi android

Berikut penjelasan kode blok pada Gambar 3.14 :

1. Jika Animasi ON, program akan mengirim data *byte* 7 pada perangkat mikrokontroler Arduino.
2. Jika Animasi OFF, program akan mengirim data *byte* 6 pada perangkat mikrokontroler Arduino.

6. Music Synched Program

Program *music synched* merupakan fitur kendali warna LED RGB berdasarkan frekuensi musik yang berputar di area sekitar sensor. Sehingga nyala LED RGB dinamis terhadap keadaan suara yang masuk dalam sensor mikrofon. Berikut *source-code* program untuk mengaktifkan fitur sinkronisasi LED RGB terhadap musik :

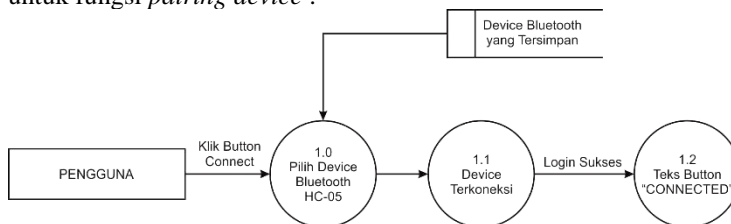


Gambar 3.15 Source-Code program *music synched* aplikasi android
Berikut penjelasan kode blok pada Gambar 3.15 :

1. Jika Musik ON, program akan mengirim data *byte* 9 pada perangkat mikrokontroler Arduino.
2. Jika Musik OFF, program akan mengirim data *byte* 8 pada perangkat mikrokontroler Arduino.

7. Bluetooth Communication

Dalam pengiriman data *byte* dari aplikasi android ke mikrokontroler Arduino dibutuhkan media transmisi, dalam hal ini digunakan media *Bluetooth* untuk melakukan komunikasi antar device. Komunikasi ini bersifat satu arah yaitu pengiriman data *byte* satu arah dari android ke mikrokontroler Arduino. Sebelum pengiriman data dilakukan, perangkat android harus disambungkan terlebih dahulu kepada modul *Bluetooth* HC-05. Berikut *source-code* program aplikasi untuk fungsi *pairing device* :

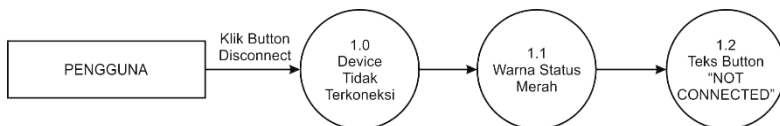


Gambar 3.16 Flowdiagram program *pairing Bluetooth device* aplikasi

Berikut penjelasan *flowdiagram* pada Gambar 3.16 :

1. Ketika button *connect* ditekan, maka akan muncul jendela pemilihan device yang tersedia. Pada saat device sudah terpilih, maka halaman kembali pada menu awal.
2. Kemudian setelah *bluetooth* sudah terkoneksi, pengguna dapat mengakses 4 fitur menu yang terdapat pada halaman utama aplikasi.
3. Mengubah status device menjadi warna hijau dan bertuliskan “CONNECTED”.

Selain mengkoneksikan device *bluetooth*, aplikasi android kendali warna LED RGB dilengkapi dengan perintah *disconnecting bluetooth*. Hal ini dilakukan apabila pengguna hendak mengubah device *Bluetooth* dengan perangkat android yang lain. Karena modul *Bluetooth* tidak bisa terkoneksi dengan 2 atau lebih jaringan. Sehingga hanya terbatas satu perangkat. Berikut instruksi program untuk *disconnecting Bluetooth* :



Gambar 3.17 *Flowdiagram disconnect device bluetooth*

Berikut penjelasan *flowdiagram* pada Gambar 3.17 :

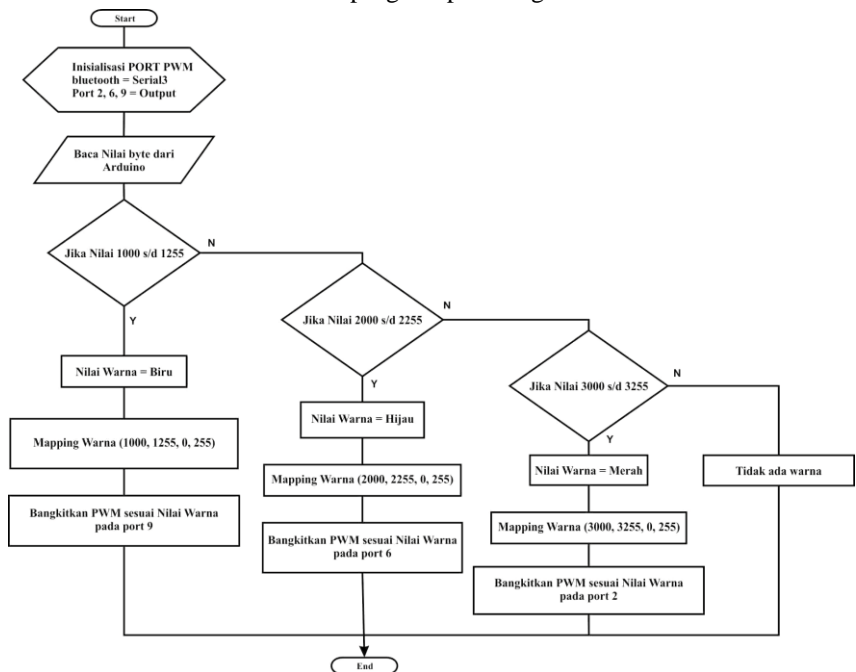
1. Ketika pengguna menekan button *disconnect*, maka aplikasi akan memproses pemutusan koneksi perangkat android dan perangkat mikrokontroler Arduino.
2. Kemudian setelah *bluetooth* sudah tidak terkoneksi, pengguna tidak dapat mengakses 4 menu yang terdapat pada halaman utama aplikasi.
3. 4 menu tersebut adalah halaman menu led1, halaman menu led2, halaman menu led3, dan halaman menu animasi.
4. Warna status device pada halaman berubah menjadi warna merah untuk menunjukkan bahwa modul sudah tidak terkoneksi.
5. Kemudian teks status berganti menjadi “NOT CONNECTED”.

3.7 Perancangan Program Arduino

Mikrokontroler dapat membuat instruksi kendali LED RGB apabila terisi program untuk mengeksekusi perintah tersebut. Pada pengendalian LED RGB terdapat 2 macam bagian pemrograman, yakni pemrograman pembangkit PWM (*Pulse Width Modulation*), program *auto-fading*, dan program *music-synched*.

3.7.1 Program Pembangkit PWM

Pada sistem pengendali warna LED RGB digunakan pembangkitan PWM dengan metode digital atau dengan menggunakan mikrokontroler. Mekanisme pembangkitan program PWM diawali dengan penerimaan perintah dari android berupa sinyal *byte* yang komunikasikan melalui *Bluetooth*. Kemudian mikrokontroler Arduino akan memroses data tersebut menjadi sebuah instruksi untuk membangkitkan *Pulse Width Modulation*. Berikut flowchart program pembangkitan PWM :



Gambar 3.18 Flowchart pembangkitan PWM

Penjelasan *flowchart* sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Inisialisasi pin PWM, pin 2 digunakan untuk PWM LED warna merah, pin 6 digunakan untuk PWM LED warna hijau, dan pin 9 digunakan untuk PWM LED warna biru. Dan Bluetooth menggunakan Serial3
3. Apabila data byte yang diterima Arduino berkisar 1000 – 1255, maka Arduino mengaktifkan PWM pada pin 9 (warna biru) dengan mapping skala 0 - 255.
4. Apabila data byte yang diterima Arduino berkisar 2000 – 2255, maka Arduino mengaktifkan PWM pada pin 6 (warna hijau) dengan mapping skala0 - 255
5. Apabila data byte yang diterima Arduino berkisar 3000 – 3255, maka Arduino mengaktifkan PWM pada pin 2 (warna merah) dengan mapping skala 0 - 255
6. Jika tidak melakukan 4 perintah diatas, maka Arduino tidak mengenali nilai warna

Sebagaimana flowchart yang telah ditampilkan pada gambar 3.18, program pembangkitan PWM akan mengatur nyala LED RGB. Bentuk eksekusi dari flowchart tersebut adalah pemrograman mikrokontroler Arduino yang dilakukan pada software Arduino IDE. Berikut segmen program pembangkitan PWM arduino :

```
else if (color >= 1000 && color <1255){
    int blueA = color;
    blueA = map(blueA, 1000,1255,0,255);
    analogWrite(5,blueA);
    Serial.println(blueA);
}
else if (color >=2000 && color <2255){
    int greenA = color;
    greenA = map(greenA,2000,2255,0,255);
    analogWrite(3,greenA);
    Serial.println(greenA);
}
else if (color >=3000 && color < 3255){
    int redA = color;
    redA = map(redA, 3000, 3255,0,255);
    analogWrite(2,redA);
    Serial.println(redA);
}
```

Gambar 3.19 Contoh segmen program pembangkitan PWM

3.7.2 Program Auto Fading

Pembuatan program *auto-fading* bertujuan untuk membuat animasi LED RGB yang akan berubah warna secara otomatis dengan waktu tertentu. Konfigurasi perubahan warna adalah dari warna merah ke hijau, kemudian dari hijau ke biru. Kondisi ini dilakukan berulang - ulang hingga sistem dihentikan oleh pengguna. Mekanisme pembangkitan program *auto-fading* diawali dengan pembangkitan penerimaan sinyal *byte* dari android kepada mikrokontroler Arduino. Kemudian Arduino akan memproses data tersebut menjadi sebuah perintah *auto-fading* LED RGB. Fitur tersebut dibangkitkan menggunakan persamaan gelombang sinusoidal :

$$Y = A * \sin((2 * \pi / T) * t)$$

Masing – masing nilai R,G, dan B di setting berbeda fasa sebesar 90 derajat sehingga respon LED akan berubah warna yang berbeda. Terdapat beberapa parameter yang dapat diubah melalui perangkat android yaitu, T (Periode) dan t (waktu tanggap). Parameter tersebut diterima dari aplikasi android dan langsung direspon oleh sistem. Berikut contoh segmentasi program *auto fading* pada arduino:

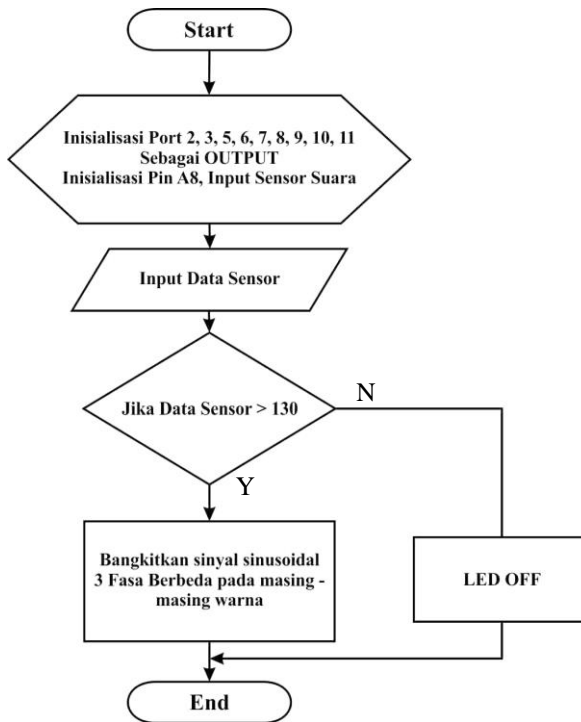
```
void rgbFading() {  
    time = millis();  
    int value1 = 255 * sin(((2 * PI / periode) * (rMax1 - time)));  
    int value2 = 255 * sin(((2 * PI / periode) * (gMax1 - time) + 90));  
    int value3 = 255 * sin(((2 * PI / periode) * (bMax1 - time) - 90));  
    int value4 = 255 * sin(((2 * PI / periode) * (rMax2 - time)));  
    int value5 = 255 * sin(((2 * PI / periode) * (gMax2 - time) + 90));  
    int value6 = 255 * sin(((2 * PI / periode) * (bMax2 - time) - 90));  
    int value7 = 255 * sin(((2 * PI / periode) * (rMax3 - time)));  
    int value8 = 255 * sin(((2 * PI / periode) * (gMax3 - time) + 90));  
    int value9 = 255 * sin(((2 * PI / periode) * (bMax3 - time) - 90));
```

Gambar 3.20 Contoh segmen program *auto-fading*

3.7.3 Program Music Synched

LED RGB dapat dinamis dan responsif dengan pengguna. Dengan fitur program *music synched* nyala LED RGB menyesuaikan diri dengan perubahan frekuensi suara yang masuk kedalam sensor mikrofon. Data yang didapat dari sensor mikrofon berupa sinyal digital akan diproses oleh mikrokontroler untuk menghasilkan perubahan warna pada LED

RGB. Berikut alur diagram pemrograman *music synched* pada mikrokontroler arduino :



Gambar 3.21 Flowchart program *music synched*

Penjelasan *flowchart* sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Inisialisasi port PWM, port A8 sebagai input sensor suara. Port 2,3,5,6,7,8,9,10,11 sebagai output LED.
3. Arduino membaca nilai analog sensor suara
4. Jika nilai sensor lebih dari 130 bangkitkan sinyal sinusoidal dengan rumus $Y = A * \sin(wt)$. Setiap port keluaran LED memiliki perbedaan fasa pada sinyal yang dibangkitkan sebesar 90 derajat.
5. Jika nilai sensor kurang dari 0 maka LED tidak menyala
6. Program berakhir (*end*).


```

musicVal = analogRead(musicSensor);
musicVal = map(musicVal, 0, 1023, 0, 255);
musicShow();
if (musicVal > 130) {
    if (redMusic > 0) {
        analogWrite(2, redMusic);
        analogWrite(3, redMusic);
        analogWrite(5, redMusic);
        delay(30);
    }
    void musicShow() {
        time = millis();
        int val1 = 255 * sin(((2 * PI / 4000) * time));
        int val2 = 255 * sin(((2 * PI / 4000) * time) + 90);
        int val3 = 255 * sin(((2 * PI / 4000) * time) - 90);
        redMusic = val1;
        greenMusic = val2;
        blueMusic = val3;
    }
}

```

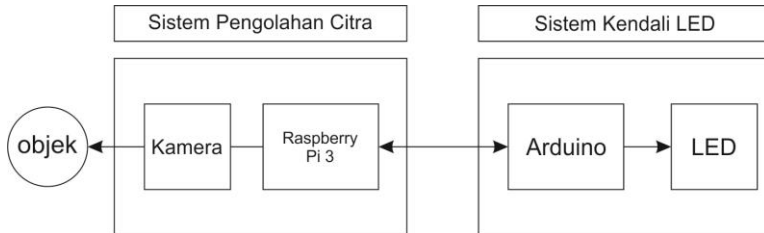
Gambar 3.22 Contoh segmen program *music synched*

3.8 Perancangan Sistem *Image Processing*

Pengolahan citra digunakan untuk mengakuisisi data RGB yang diperoleh dari hasil rekam objek. Sensor kamera akan menangkap visual objek kemudian data tersebut diolah oleh raspberry pi. Keluaran pengolahan citra yang dilakukan adalah data nilai warna RGB objek tersebut. Sistem pengolahan citra difokuskan pada raspberry pi saja lalu skema pengendalian warna LED akan dieksekusi oleh mikrokontroler arduino. Arduino akan menerima data RGB dari raspberry pi dan kemudian melakukan perintah untuk membangkitkan PWM pin warna LED RGB.

3.8.1 Blok Diagram Sistem *Image Processing*

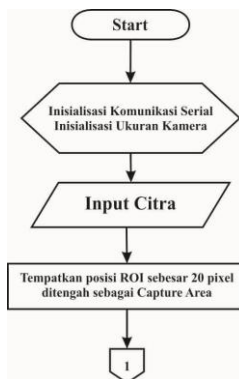
Berdasarkan pada rancangan sistem diatas, blok sistem *image processing* yang digunakan untuk implementasi alat kendali LED RGB adalah seperti dibawah ini :

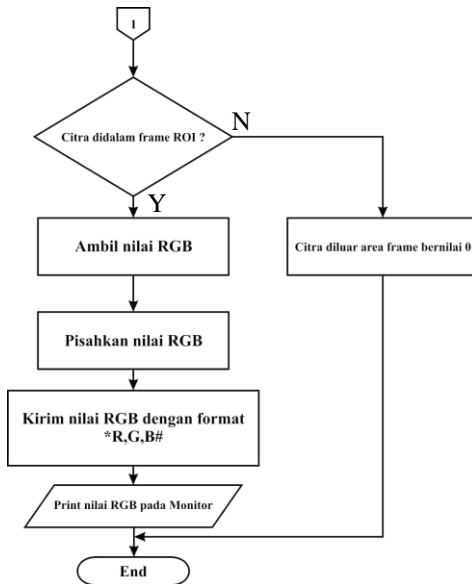


Gambar 3.23 Blok diagram sistem pengolahan citra

Berdasarkan gambar 3.23 blok diagram sistem, Raspberry Pi akan mengolah gambar yang telah diambil oleh kamera. Kamera dipasang pada objek dalam hal ini adalah LED RGB. Kamera diberi filter berupa kertas yang ditempel pada lensa, sehingga kamera dapat menangkap citra dengan baik. Hasil dari pengolahan tersebut berupa data nilai RGB. Selanjutnya nilai akan dikirim kepada mikrokontroler Arduino melalui komunikasi serial. Komunikasi serial yang digunakan adalah serial USB. Arduino menangkap data tersebut dalam bentuk 3 nilai. Sehingga dibutuhkan program pemecahan data (*parsing data*) untuk mengidentifikasi masing – masing nilai R, G, dan B. Sebagaimana pada tujuan membuat sistem pengolahan citra adalah untuk mengenali objek melalui kamera. Kemudian sistem akan menentukan nilai warna. Arduino sebagai mikrokontroler membangkitkan warna sesuai dengan warna yang dikenali pada kamera.

3.8.2 Diagram Alir Program *Image Processing*





Gambar 3.24 Flowchart Program Pengolahan Citra

Penjelasan *flowchart* sebagai berikut :

1. *Start* adalah ketika program dimulai.
2. Inisialisasi komunikasi serial, memanggil fungsi – fungsi komunikasi serial. Setting ukuran resolusi kamera 1360 * 720 pixel
3. Raspberry mengaktifkan kamera dan mendapatkan nilai RGB.
4. Sistem mengakuisisi data berdasarkan ROI (*Range of Interest*) yakni area yang akan diambil citranya.
5. Jika citra didalam ROI maka ambil data RGB. Jika tidak nilai warna = 0
6. Data yang sudah didapatkan berupa data string[3]. Maka program harus memisahkan masing – masing nilai R, G, dan B.
7. Mengatur format nilai RGB menjadi *R,G,B#, Kemudian Kirim data tersebut pada arduino.
8. Cetak data RGB pada monitor.
9. Program berakhir (*end*).

3.8.3 Komunikasi Raspberry Pi dengan Arduino

Pada sistem pengolahan citra, hasil data yang diperoleh berupa data warna RGB dengan format *R,G,B#. Kemudian data tersebut akan dikirim pada perangkat arduino sebagai masukan sensor warna melalui komunikasi serial (kabel). Selanjutnya arduino memproses data masukan tersebut untuk memberikan perintah pembangkitan pwm. Namun untuk mendefinisikan data tersebut dibutuhkan proses *parsing data* yakni mengidentifikasi masing – masing data red, green, dan blue. Arduino akan mengenali tiap komponen warna melalui format data yang dikirim oleh raspberry pi. Warna merah dikenali dengan format *R, warna hijau dikenali dengan format G, dan warna biru dapat dikenali dengan format B#. Setelah itu arduino dapat memproses nilai sensor warna untuk membuat sebuah program kendali LED warna RGB. Berikut segmentasi program untuk memecah data 3 warna :

```
void parsingData() {  
    int j = 0;  
    dt[j] = "";  
    for (i = 1; i < dataIn.length(); i++) {  
        if ((dataIn[i] == '#' || (dataIn[i] == ',')) {  
            j++;  
            dt[j] = "";  
        }  
        else {  
            dt[j] = dt[j] + dataIn[i];  
        }  
    }  
    Rx = dt[0].toInt();  
    Gx = dt[1].toInt();  
    Bx = dt[2].toInt();  
}
```

Gambar 3.25 Segmentasi Program *Parsing Data Image Processing*

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

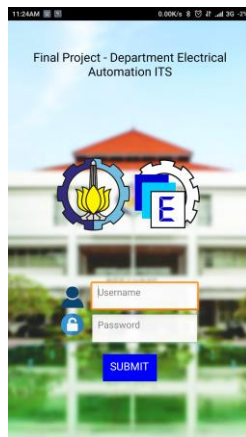
Bab ini dibagi menjadi empat bagian. Pada bagian pertama dijelaskan mengenai implementasi antarmuka aplikasi, bagian kedua dijelaskan hasil pengujian antarmuka aplikasi, bagian ketiga dijelaskan pengujian sistem pengolahan citra, dan bagian terakhir dijelaskan pengujian sistem secara menyeluruh.

4.1 Implementasi Antarmuka Aplikasi Android

Berdasarkan perancangan *software* yang telah dijelaskan pada bab III aplikasi yang telah dibuat dengan app inventor didownload dengan cara Build App (save .apk to my computer). Setelah file berhasil terdownload, file aplikasi yang berbentuk .apk diinstall pada perangkat android. Aplikasi android terbagi menjadi 4 bentuk halaman yaitu halaman login, halaman menu, halaman led, dan halaman animasi.

4.1.1 Halaman Login

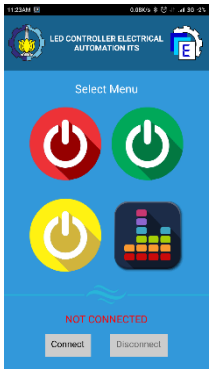
Halaman login merupakan halam pertama yang muncul pada aplikasi. Pada halaman login pengguna harus memasukkan *username* dan *password* untuk mengakses halaman selanjutnya (halaman menu). Berikut implementasi halaman login :



Gambar 4.1 Halaman login aplikasi android

4.1.2 Halaman Menu

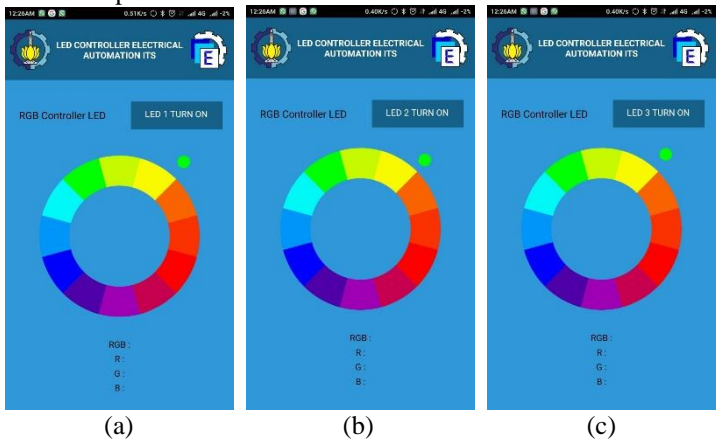
Halaman menu adalah halaman utama pada aplikasi android. Bertujuan untuk memilih fitur menu yang ada didalamnya. Terdapat 4 menu yakni Led1, Led2, Led3, Animasi. Berikut implementasi halaman menu :



Gambar 4.2 Halaman menu aplikasi android

4.1.3 Halaman LED

Berikut implementasi halaman LED :

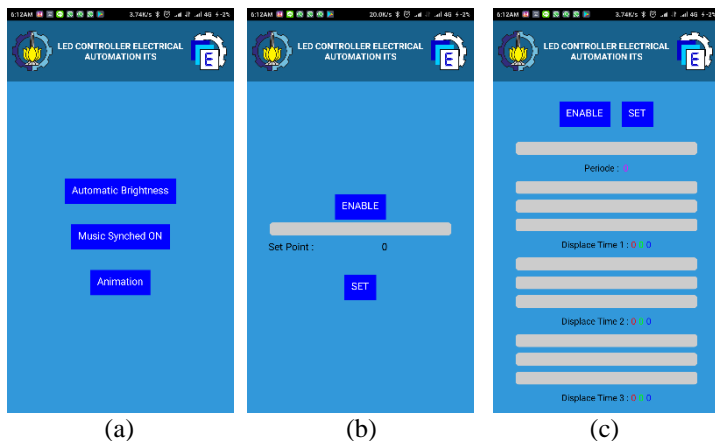


Gambar 4.3 (a) Halaman kontrol led 1 (b) Halaman kontrol led 2 (c) Halaman kontrol led 3

Halaman led terdapat 3 buah yaitu led1, led2, dan led3 karena LED RGB yang dikendalikan berjumlah 3 buah. Halaman led berfungsi sebagai pengendalian warna dan nyala/mati LED RGB. Untuk mengatur nyala/mati led dapat dilakukan dengan tombol LED TURN ON. Dan untuk mengatur warna LED RGB dapat diatur dengan palet warna.

4.1.4 Halaman Animasi

LED RGB dilengkapi dengan fitur *auto-brightness*, animasi *auto-fading* dan *music synched* dimana pengendalian 3 fitur tersebut diatur pada halaman animasi. Berikut implementasi halaman animasi :



Gambar 4.4 (a) Halaman kontrol animasi (b) Halaman kontrol *auto-brightness* (c) Halaman kontrol *auto-fading*

4.2 Pengujian Antarmuka Aplikasi Android

Pengujian aplikasi android memiliki tujuan untuk menguji keterhubungan antara desain aplikasi dan fungsi – fungsi program kode blok yang telah diimplementasikan dalam aplikasi. Dengan pengujian ini dapat diketahui kinerja sistem aplikasi dalam mengeksekusi instruksi program yang telah dibuat.

4.2.1 Pengujian Antarmuka Login

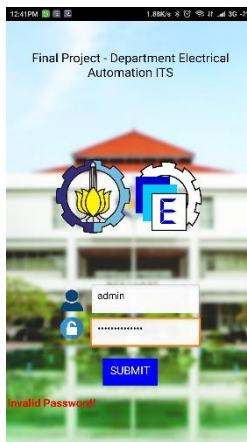
Dalam pengujian antarmuka login terdapat 2 skenario pengujian. Pertama menguji halaman login ketika username dan password benar.

Jika benar, maka pada aplikasi akan berpindah halaman menjadi halaman menu.



Gambar 4.5 Antarmuka login aplikasi android

Terdapat textbox username dan textbox password pada halaman login. Jika username “admin” dan password “akusayangkamU”, screen berpindah ke halaman menu. Jika admin dan password salah maka muncul *message-alert* bertuliskan “Invalid Password!”. Hal ini menandakan bahwa login gagal. Berikut uji coba login gagal :




Gambar 4.6 Antarmuka pengujian login gagal

Dari pengujian login benar dan login salah, antarmuka berhasil mengeksekusi perintah sesuai fungsi kode blok yang telah diprogram.

4.2.2 Pengujian Antarmuka Menu

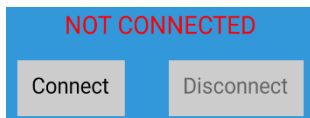
Berikut hasil pengujian antarmuka menu :

Tabel 4.1 Pengujian Antarmuka Menu Area

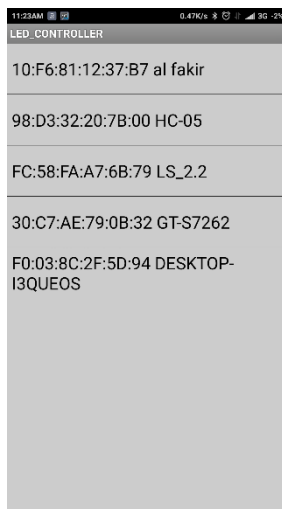
Skenario Pengujian	Tombol Button	Halaman Tujuan	Kesimpulan Pengujian
Button led1 diklik, kemudian berpindah ke halaman Led1.			Berhasil
Button led2 diklik, kemudian berpindah ke halaman Led1.			Berhasil
Button led3 diklik, kemudian berpindah ke halaman Led3.			Berhasil
Button animasi diklik, kemudian berpindah ke halaman animasi.			Berhasil

4.2.3 Pengujian *Pairing Bluetooth Device*

Percobaan ini dilakukan untuk menguji konektivitas Bluetooth dengan aplikasi android. Skenario uji cobanya adalah pada menu utama ketika button connect ditekan, maka muncul jendela pemilihan perangkat Bluetooth yang tersedia.

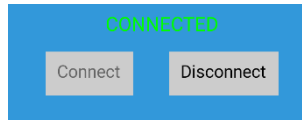


Gambar 4.7 Antarmuka connect bluetooth



Gambar 4.8 Jendela listview device bluetooth

Pada saat sebelum device terkoneksi, status pada aplikasi adalah "NOT CONNECTED". Kemudian setelah tombol connect ditekan muncul jendela listview device Bluetooth yang tersedia. Untuk alamat modul Bluetooth yang tersambung pada mikrokontroler adalah "98:D3:32:20:7B:00 HC-05". Ketika sudah *terpaired* dengan perangkat Bluetooth, Kemudian status button connect menjadi disable, button disconnected menjadi enable, dan teks menjadi "CONNECTED".



Gambar 4.9 Antarmuka disconnect bluetooth

4.2.4 Pengujian Kendali Switch LED

Pengujian kendali LED memiliki tujuan untuk menguji fungsi kendali button dan palet warna. Pengaturan led terdapat pada halaman led. Kendali button pada setiap LED RGB difungsikan sebagai kontrol nyala/mati LED RGB dengan warna default yaitu putih. Sedangkan kendali *color picker* berfungsi untuk mengatur warna berdasarkan pemilihan warna oleh pengguna pada palet warna. Melalui perintah tersebut, aplikasi android berperan sebagai pengendali dari mikrokontroler arduino. Aplikasi akan mengirim data dalam bentuk aplikasi android berperan sebagai pengendali dari mikrokontroler arduino. Aplikasi akan mengirim data dalam bentuk *byte* melalui komunikasi Bluetooth. Mikrokontroler menerima data tersebut dan akan memproses data menjadi program pembangkitan PWM. Berikut uji coba data yang dikirim melalui kendali LED :

Tabel 4.2 Data pengiriman kendali LED

Halaman	User Interface	Data byte yang dikirim	Data byte yang diterima	Status
LedScreen1	Button LED1 Switch OFF	0	0	Berhasil
	Button LED1 Switch ON	1	1	Berhasil
LedScreen2	Button LED2 Switch OFF	2	2	Berhasil
	Button LED2 Switch ON	3	3	Berhasil
LedScreen3	Button LED3 Switch OFF	4	4	Berhasil
	Button LED3 Switch ON	5	5	Berhasil

Dari pengujian yang dilakukan informasi data yang dikirim dan diterima, sistem berhasil melakukan komunikasi antara antarmuka aplikasi sebagai pengirim data dan perangkat mikrokontroler arduino sebagai penerima data.

4.2.5 Pengujian Kendali Animasi

Pengujian ini dimaksudkan untuk menguji fungsi kendali animasi oleh antarmuka aplikasi. Kendali ini berupa pengiriman data *byte* oleh aplikasi pada halaman animasi untuk mengaktifkan fitur *auto-fading* dan *music synched*. Sehingga mikrokontroler dapat memproses data yang diterima untuk mengaktifkan fitur tersebut. Berikut data uji coba yang dikirim melalui kendali animasi :

Tabel 4.3 Data pengiriman kendali animasi

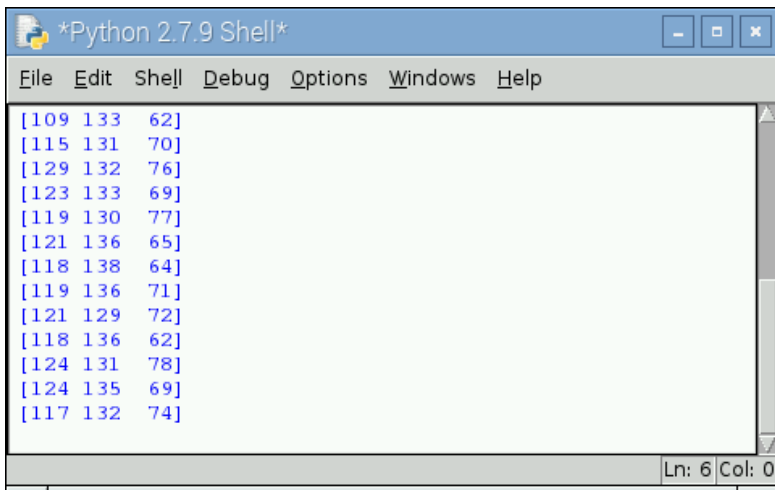
Halaman	User Interface	Data byte yang dikirim	Data byte yang diterima	Status
Animation Screen	Button Animation OFF	6	6	Berhasil
	Button Animation ON	7	7	Berhasil
	Button Music Synched OFF	8	8	Berhasil
	Button Music Synched ON	9	9	Berhasil
	Automatic Mode OFF	10	10	Berhasil
	Automatic Mode ON	11	11	Berhasil

Berdasarkan uji coba yang dilakukan, data yang dikirim oleh aplikasi dapat secara tepat diterima oleh mikrokontroler arduino,

sehingga antara aplikasi perangkat android dan mikrokontroler arduino dapat berkomunikasi dengan baik.

4.3 Pengujian Sistem *Image Processing*

Pengujian sistem pengolahan citra bertujuan untuk mengkomunikasikan perangkat raspberry pi dengan perangkat arduino. Raspberry pi mengirimkan data nilai R, G, B pada perangkat android melalui komunikasi serial USB. Berikut hasil pengiriman data Raspberry pi :

A screenshot of a Python 2.7.9 Shell window. The window has a title bar with the text '*Python 2.7.9 Shell*' and standard window control buttons (minimize, maximize, close). Below the title bar is a menu bar with options: File, Edit, Shell, Debug, Options, Windows, and Help. The main area of the window is a text editor displaying a list of 15 lines of data, each representing an RGB value in the format [R G B]. The data is as follows:

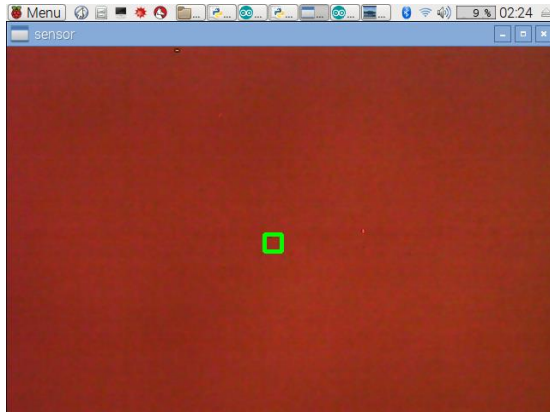
[109	133	62]
[115	131	70]
[129	132	76]
[123	133	69]
[119	130	77]
[121	136	65]
[118	138	64]
[119	136	71]
[121	129	72]
[118	136	62]
[124	131	78]
[124	135	69]
[117	132	74]

The status bar at the bottom right of the window shows 'Ln: 6 Col: 0'.

Gambar 4.10 Pengiriman data RGB Raspberry pi

Pada gambar 4.10 raspberry pi melalui aplikasi python 2.7.9 berhasil mengirim data pada perangkat arduino melalui serial USB. Data yang dikirimkan berdasarkan masukan dari *webcam*. Data *default* yang olah raspberry pi memiliki format [b g r].

Berdasarkan gambar 4.10 data yang dicetak pada layar monitor r, g, b adalah data yang akan dikirimkan kepada arduino. Dan data warna yang berformat [b g r] adalah data mentah yang didapatkan dari *webcam* (*web camera*). Berikut pengambilan data warna oleh *Webcam* :



Gambar 4.11 Pengambilan data warna dengan Webcam

Pada gambar 4.11 ukuran kamera yang akuisisi adalah 480 x 320 pixel. Namun data diambil hanya pada area dalam kotak warna hijau. Area ROI (*Range of Interest*) tersebut berukuran 10 x 10 pixel. Selanjutnya data dikirim pada perangkat android dengan format *R,G,B#. Pengiriman data melalui komunikasi serial USB dengan kecepatan *baudrate* 9600 bps. Setelah mikrokontroler arduino menerima data, arduino melakukan pemecahan data *string* rgb (*parsing data*). Berikut hasil penerimaan data rgb :

```

/dev/ttyACM0 (Arduino/Genuino Mega or Mega 2560)

data G : 153
data B : 155

data masuk : * 94,156,146#

data R : 94
data G : 156
data B : 146

data masuk : * 91,156,152#

data R : 91
data G : 156
data B : 152
  
```

Gambar 4.12 Program *Parsing Data* Arduino

Data warna RGB berhasil diterima oleh arduino. Pada gambar 4.12 data masuk yang berformat *R,G,B#. Arduino kemudian melakukan pemecahan data untuk mengenali masing – masing nilai warna merah, hijau dan biru. Warna merah ditandai dengan *R, warna hijau ditandai dengan G, dan warna biru ditandai dengan B#. Kemudian masing – masing nilai warna tersebut disimpan pada variabel Rx, Gx, dan Bx dan arduino melakukan perintah pembangkitan PWM berdasarkan nilai Rx, Gx, dan Bx sehingga warna objek yang di kenali oleh kamera sama dengan warna LED.

4.4 Pengujian Keseluruhan Sistem

Perkembangan Pengujian sistem ini memiliki tujuan untuk menguji kinerja serta hubungan antara perangkat keras dengan perangkat lunak sebagai program aplikasis system. Dengan pengujian ini dapat diketahui apakah alat dan aplikasi yang telah dirancang dapat bekerja sesuai dengan yang diinginkan atau tidak. Cara pengujian ini dilakukan dalam pengujian perangkat keras dan perangkat lunak. Pengujian kali ini bertujuan untuk mengetahui kemampuan dari Arduino dalam memproses perintah yang masuk dari Android dengan jarak kontrol yang berbeda – beda.

Tabel 4.4 Pengujian jarak kontrol dan waktu tanggap bluetooth

Percobaan ke-	Jarak Kontrol (meter)	Konektivitas	Waktu Tanggap (detik)
1	1 (indoor)	Terkoneksi	0.1
2	2 (indoor)	Terkoneksi	0.2
3	3 (indoor)	Terkoneksi	0.3
4	5 (indoor)	Terkoneksi	0.5
5	7 (indoor)	Terkoneksi	1
6	8 (indoor)	Terkoneksi	1
7	10 (indoor)	Terkoneksi	5>
8	15 (indoor)	Tidak Terkoneksi	-
9	20 (indoor)	Tidak Terkoneksi	-

Dari pengujian yang dilakukan dapat dilihat performa dari Arduino dalam menerima perintah serta mengirimkan *feedback* ke Android menggunakan bluetooth. Waktu yang diperlukan oleh arduino untuk mengeksekusi perintah yang diterima berkisar antara 0.5 sampai 3 detik.

Kecepatan tanggap dari Arduino dipengaruhi oleh jarak serta halangan dan kekuatan sinyal dari bluetooth yang digunakan, semakin baik kualitas dari bluetooth maka semakin jauh jarak koneksi untuk melakukan pengiriman pengiriman ke Arduino

Pengujian tahap selanjutnya adalah pengujian warna LED RGB dan beberapa fitur pendukung. Pengujian ini dilakukan dengan 4 skenario yaitu :

1. Pengujian switch ON/OFF LED RGB.
2. Pengujian warna LED RGB berdasarkan warna palet yang disentuh.
3. Pengujian animasi *auto-fading* LED RGB.
4. Pengujian animasi *music-synced* LED RGB.

Pengujian dilakukan dengan melibatkan perangkat keras yang terhubung pada mikrokontroler arduino mega. Hasil yang diharapkan adalah mikrokontroler arduino dapat menerima perintah dan mengeksekusi perintah tersebut dalam bentuk pembangkitan PWM. Adapun jumlah PWM yang dibangkitkan adalah 9 pin. Setiap pin merepresentasikan satu warna pada LED RGB. Sehingga terdapat 3 LED RGB yang akan di uji coba.

1. Pengujian Kendali Switch ON/OFF LED RGB

Tabel 4.5 Pengujian nyala/mati LED RGB

Skenario	Data Kendali yang dikirim	Hasil yang diharapkan	Hasil pengujian LED	Nilai PWM
Button ON LED 1 ditekan	1	LED pertama menyala putih	ON	R = 255 G = 255 B = 255
Button ON LED 2 ditekan	3	LED kedua menyala putih	ON	
Button ON LED 3 ditekan	5	LED ketiga menyala putih	ON	

Button OFF LED 1 ditekan	0	LED pertama mati	OFF	R = 0 G = 0 B = 0
Button OFF LED 2 ditekan	2	LED kedua mati	OFF	
Button OFF LED 3 ditekan	4	LED ketiga mati	OFF	

Berdasarkan tabel 4.5 hasil pengujian kendali LED RGB dengan perangkat android berhasil dilakukan, hal ini dibuktikan dengan perubahan LED yang semula mati menjadi menyala pada saat button switch pada aplikasi android ditekan. Dan antara LED1, LED2, dan LED3 menyala tidak bersamaan sehingga LED dapat dikontrol satu persatu.

2. Pengujian Kendali Warna LED RGB

Pengujian warna LED RGB bertujuan untuk mengetahui perubahan warna pada LED RGB yang dikendalikan melalui aplikasi android. Berikut data RGB yang diberikan pada uji coba :

Tabel 4.6 Pengujian warna LED RGB

No	Data Input R	Data Input G	Data Input B	Data Sensor R	Data Sensor G	Data Sensor B	Rata Rata Error
1	255	255	255	250	255	246	14.33
2	255	0	0	255	41	31	28.83
3	200	0	80	241	48	92	36.17

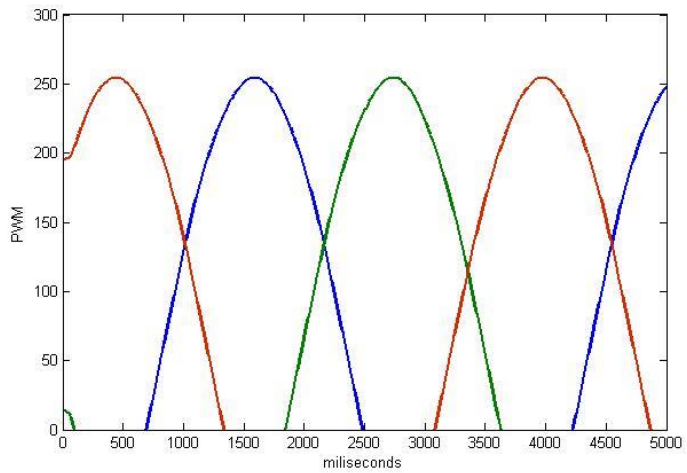
4	160	1	180	204	55	162	33.17
5	80	1	170	106	41	153	28
6	0	1	255	37	20	226	19.17
7	1	151	250	0	144	228	12
8	1	251	250	0	220	240	23
9	1	255	0	0	206	46	32.67
10	201	250	0	180	234	63	26.33
11	250	100	0	245	110	43	20.17
12	250	50	0	255	82	26	21

Dari pengujian yang dilakukan dapat dilihat performa kecerahan LED RGB yang di uji dari setiap nilai *setpoint* yang diberikan oleh perangkat android menghasilkan data masukan sensor dengan nilai rata rata tidak lebih dari 37. Pada saat nilai input PWM dikirim dari android ke arduino, terjadi perubahan warna LED mengikuti dengan nilai PWM yang diberikan. Terdapat 12 macam kombinasi warna yang dapat dipilih oleh pengguna. Masing – masing kombinasi warna memiliki nilai antara 0 – 255. Nilai ini linier dengan nilai PWM. Selanjutnya nilai sensor akan di evaluasi oleh sistem untuk membuat perintah agar nilai error = 0 (*zero offset*). Sehingga yang diharapkan adalah nilai *setpoint* dan nilai sensor sama.

3. Pengujian Animasi *auto-fading* LED RGB

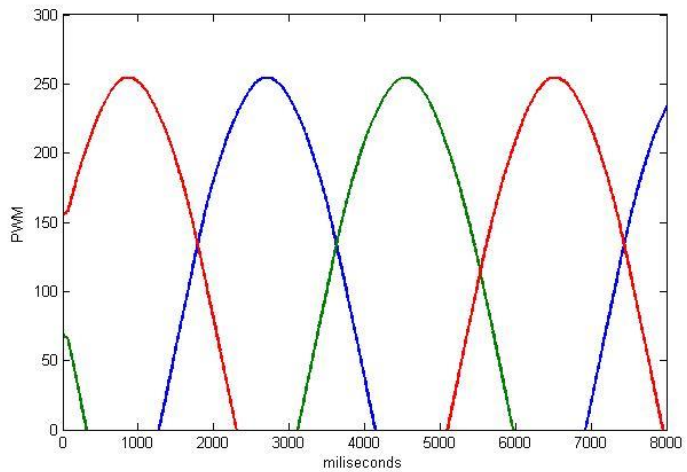
Pengujian animasi ini memiliki tujuan untuk menguji fungsi perubahan warna LED RGB dari suatu warna ke warna lainnya dengan ketentuan waktu tertentu. Perubahan warna tersebut akan berjalan otomatis setelah aplikasi mengirim perintah aktivasi program. Skenario pengujian adalah melihat respon perubahan warna dan mengukur nilai PWM saat periode tertentu sehingga mendapatkan data hasil pengujian program. Berikut hasil pengujian animasi *auto-fading* LED RGB :

- a. Set Periode = 5 detik, sample data = 5000 ms.



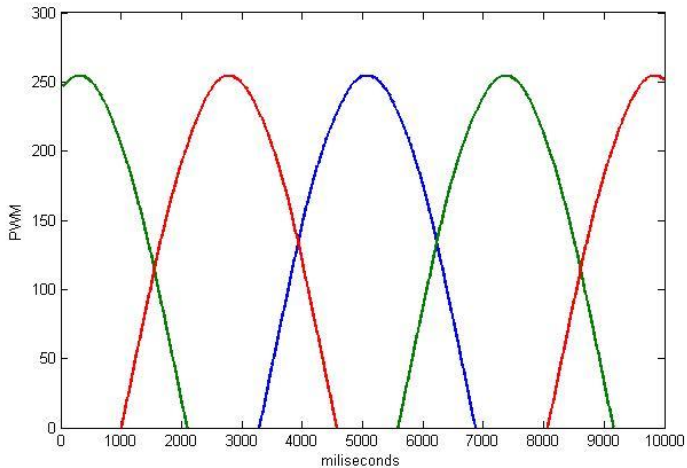
Gambar 4.13 Grafik respon animasi LED dengan periode 5 detik

- b. Set Periode = 8 detik, sample data = 8000 ms.



Gambar 4.14 Grafik respon animasi LED dengan periode 8 detik

- c. Set Periode = 10 detik, sample data = 10000 ms.



Gambar 4.15 Grafik respon animasi LED dengan periode 10 detik

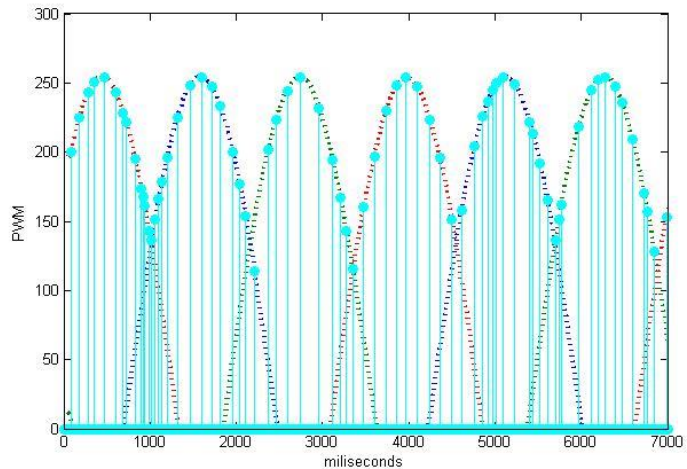
Berdasarkan gambar 4.12, 4.14, dan 4.15 respon LED dalam program animasi berbentuk sinyal sinusoidal yang dibangkitkan pada port PWM arduino. Pada perangkat android, parameter periode dapat di atur sesuai keinginan dari skala 0 – 10 detik. Kemudian parameter tersebut dikirim pada arduino. Arduino akan merespon perubahan parameter. Pada periode 5, sinyal dapat mencapai 1 gelombang dalam 5 detik. Pada saat periode 8, sinyal dapat mencapai 1 gelombang dalam 8 detik, dan Pada periode 10, sinyal mencapai 1 gelombang dalam waktu 10 detik. Hal ini dibuktikan dari tiap sampel mempunyai timer 1 *milliseconds*. Pencapaian 1 gelombang sinus dibuktikan dengan adanya 1 bukit dan 1 lembah. Nilai PWM < 0 tidak di bangkitkan karena jika dibangkitkan perubahan warna LED RGB tidak bervariasi.

4. Pengujian Animasi *music-synched* LED RGB

Pengujian animasi music bertujuan untuk mengetahui respon LED warna terhadap sinyal masukan suara dari sensor. Pengujian ini dilakukan dengan cara menyalakan musik dan didekatkan pada mikrofon sensor suara. Kemudian sensor suara dikalibrasi sensitivitasnya dan hasil data pembacaan analog di tampilkan pada serial

monitor arduino. Data respon dari masing – masing LED juga ditampilkan pada serial monitor. Berikut hasil responnya :

1. Pengujian data jika sensor suara > 130. Berikut grafik respon LED selama 7 detik.



Gambar 4.16 Grafik respon LED pada program *music-synched*

Berdasarkan hasil pengujian yang telah dilakukan, sinyal input dari sensor suara berbentuk sinyal analog. Sinyal analog memiliki rentang nilai 0 – 1023 kemudian diubah menjadi skala 0 – 255. Arduino terlebih dahulu membangkitkan sinyal sinusoidal pada masing – masing LED yang berbeda fasa sebesar 90 derajat. Pada saat sensor suara memberikan nilai > 130, Arduino akan membangkitkan nilai pwm pada saat momen t (*milliseconds*). Pada respon sinyal gambar 4.16 saat $t = 80$ ms, nilai PWM = 200. Maka sistem akan merespon dan membangkitkan LED RGB dengan nilai 200.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Dari pengujian sistem kontrol hasil desain dengan simulasi maupun implementasi, aplikasi sistem kendali LED RGB sudah direalisasikan dan berhasil dioperasikan dengan baik berdasarkan fungsi – fungsi yang telah dibuat. Aplikasi dapat mengendalikan warna LED RGB serta dapat mengakses fitur *auto brightness*, animasi *auto-fading*, dan *music synched*. Proses pengiriman data *byte* dari android ke arduino melalui modul *Bluetooth* bekerja optimal dengan tingkat akurasi 95.21%, namun terbatas jarak maksimal 9 - 10 meter. Sistem pengolahan citra menggunakan Raspberry Pi berhasil di implementasikan dengan hasil warna LED dapat berubah sesuai dengan warna objek yang ditangkap oleh kamera. Sehingga dapat disimpulkan bahwa aplikasi kendali warna LED RGB berbasis android berhasil mengendalikan warna LED RGB sesuai dengan mode dan fitur yang telah dirancang. Mode dan fitur tersebut terangkai menjadi satu tampilan antarmuka yang baik dan dinamis.

Saran untuk penelitian selanjutnya adalah mengembangkan perluasan jarak kontrol yang tidak hanya dalam ruang lingkup lokal namun sudah bisa menggunakan internet sebagai media komunikasi dengan membuat sebuah *website* yang berfungsi sebagai jembatan data antara Android dan Arduino. Menambahkan fitur – fitur animasi lainnya agar sistem kendali lebih bervariasi. Serta menggunakan program *filtering* pada sistem pengolahan citra agar menghasilkan data yang lebih akurat.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Akima, Radius. [www.repository.amikom.ac.id/files/ Publikasi_10.11.3603.pdf](http://www.repository.amikom.ac.id/files/Publikasi_10.11.3603.pdf). "*Perancangan Aplikasi dan Instalasi Kontrol Kipas Angin Berbasis Android*", Yogyakarta, 2015
- [2] Datasheet *Microcontroller* ATmega 2560.
- [3] Datasheet Transistor Mosfet IRF 540.
- [4] Farizan, Jazman. 2016. "*Kunci Pintu Otomatis dengan Kombinasi RGB Menggunakan Smartphone Android*", Yogyakarta.
- [5] MIT App Inventor. "About App Inventor". <http://appinventor.mit.edu/explore/aboutus.html>. (diakses tanggal 20 Maret 2017).
- [6] Mulyana, Eneung. "*App Inventor : Ciptakan Sendiri Aplikasi Androidmu*", Andi Publisher, Yogyakarta, 2012.
- [7] Pujiono, "*Rangkaian Listrik*", Graha Ilmu, Yogyakarta, 2013.
- [8] Rumopa, Vernando W. http://repository.polimdo.ac.id/449/1/Vernando_Waldi_Rumopa.pdf. "*Kontrol Penerangan Ruangan Menggunakan Sensor Suara (Speech Recognition) Berbasis Android*", Manado, 2015.
- [9] Setiawan, Soim. <https://www.scribd.com/doc/256336645/Tugas-Akhir>. "*Miniatur Pengendali Pintu Gerbang Menggunakan Smartphone Via Bluetooth Berbasis Arduino Uno*", Jakarta, 2014.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

TABEL HASIL PENGUJIAN

Tabel 4.7 Pengujian komunikasi *interface* android dengan arduino

No	Perintah <i>Interface</i>	Data yang dikirim	Data yang diterima
1	Switch Off LED 1	0	0
2	Switch On LED 1	1	1
3	Switch Off LED 2	2	2
4	Switch On LED 2	3	3
5	Switch Off LED 3	4	4
6	Switch On LED 3	5	5
7	Mode <i>Auto Fading</i> Off	6	6
8	Mode <i>Auto Fading</i> On	7	7
9	Mode <i>Music Synched</i> Off	8	8
10	Mode <i>Music Synched</i> On	9	9
11	Mode <i>Auto Brightness</i> Off	10	10
12	Mode <i>Auto Brightness</i> On	11	11
13	<i>Set Animation</i> Parameter	13	13
14	<i>Set Setpoint</i> Parameter	14	14
Presentase Keberhasilan			100 %

Tabel 4.8 Pengujian Komunikasi Data LED 1 Warna Merah

No	Data yang Dikirim	Data yang Diterima	Status
1	1000	1000	Berhasil
2	1005	1005	Berhasil
3	1010	1010	Berhasil
4	1015	1015	Berhasil
5	1020	1020	Berhasil
6	1025	1025	Berhasil
7	1030	1030	Berhasil
8	1035	1035	Berhasil
9	1040	1040	Berhasil
10	1045	1045	Berhasil
11	1050	1050	Berhasil
12	1055	-	Loss
13	1060	1060	Berhasil
14	1065	1065	Berhasil
15	1070	1070	Berhasil
16	1075	-	Loss
17	1080	-	Loss
18	1090	1090	Berhasil
19	1100	1100	Berhasil
20	1110	1110	Berhasil
21	1120	1120	Berhasil
22	1130	1130	Berhasil
23	1140	1140	Berhasil
24	1150	1150	Berhasil
25	1160	-	Loss
26	1170	-	Loss
27	1180	1180	Berhasil
28	1190	1190	Berhasil
29	1200	1200	Berhasil
30	1210	1210	Berhasil
31	1220	1220	Berhasil
32	1230	1230	Berhasil
33	1240	1240	Berhasil
34	1255	1255	Berhasil
Presentase Keberhasilan			85.3%

Tabel 4.9 Pengujian Komunikasi Data LED 1 Warna Hijau

No	Data yang Dikirim	Data yang Diterima	Status
1	2000	2000	Berhasil
2	2005	-	Loss
3	2010	2010	Berhasil
4	2015	-	Loss
5	2020	2020	Berhasil
6	2025	2025	Berhasil
7	2030	2030	Berhasil
8	2035	2035	Berhasil
9	2040	2040	Berhasil
10	2045	2045	Berhasil
11	2050	2050	Berhasil
12	2055	2055	Berhasil
13	2060	2060	Berhasil
14	2065	2065	Berhasil
15	2070	2070	Berhasil
16	2075	2075	Berhasil
17	2080	2080	Berhasil
18	2090	2090	Berhasil
19	2100	2100	Berhasil
20	2110	2110	Berhasil
21	2120	2120	Berhasil
22	2130	2130	Berhasil
23	2140	2140	Berhasil
24	2150	2150	Berhasil
25	2160	2160	Berhasil
26	2170	2170	Berhasil
27	2180	2180	Berhasil
28	2190	2190	Berhasil
29	2200	2200	Berhasil
30	2210	2210	Berhasil
31	2220	2220	Berhasil
32	2230	2230	Berhasil
33	2240	2240	Berhasil
34	2255	2255	Berhasil
Presentase Keberhasilan			94.2%

Tabel 4.10 Pengujian Komunikasi Data LED 1 Warna Biru

No	Data yang Dikirim	Data yang Diterima	Status
1	3000	3000	Berhasil
2	3005	3005	Berhasil
3	3010	3010	Berhasil
4	3015	3015	Berhasil
5	3020	3020	Berhasil
6	3025	3025	Berhasil
7	3030	3030	Berhasil
8	3035	3035	Berhasil
9	3040	3040	Berhasil
10	3045	3045	Berhasil
11	3050	3050	Berhasil
12	3055	3055	Berhasil
13	3060	3060	Berhasil
14	3065	3065	Berhasil
15	3070	3070	Berhasil
16	3075	3075	Berhasil
17	3080	3080	Berhasil
18	3090	3090	Berhasil
19	3100	3100	Berhasil
20	3110	3110	Berhasil
21	3120	3120	Berhasil
22	3130	3130	Berhasil
23	3140	3140	Berhasil
24	3150	3150	Berhasil
25	3160	3160	Berhasil
26	3170	3170	Berhasil
27	3180	3180	Berhasil
28	3190	3190	Berhasil
29	3200	3200	Berhasil
30	3210	3210	Berhasil
31	3220	3220	Berhasil
32	3230	3230	Berhasil
33	3240	3240	Berhasil
34	3255	3255	Berhasil
Presentase Keberhasilan			100%

Tabel 4.11 Pengujian Komunikasi Data LED 2 Warna Merah

No	Data yang Dikirim	Data yang Diterima	Status
1	4000	4000	Berhasil
2	4005	4005	Berhasil
3	4010	4010	Berhasil
4	4015	4015	Berhasil
5	4020	4020	Berhasil
6	4025	4025	Berhasil
7	4030	4030	Berhasil
8	4035	4035	Berhasil
9	4040	4040	Berhasil
10	4045	4045	Berhasil
11	4050	4050	Berhasil
12	4055	4055	Berhasil
13	4060	4060	Berhasil
14	4065	4065	Berhasil
15	4070	4070	Berhasil
16	4075	-	Loss
17	4080	4080	Berhasil
18	4090	4090	Berhasil
19	4100	4100	Berhasil
20	4110	4110	Berhasil
21	4120	4120	Berhasil
22	4130	4130	Berhasil
23	4140	4140	Berhasil
24	4150	4150	Berhasil
25	4160	-	Loss
26	4170	4170	Berhasil
27	4180	4180	Berhasil
28	4190	4190	Berhasil
29	4200	4200	Berhasil
30	4210	4210	Berhasil
31	4220	4220	Berhasil
32	4230	4230	Berhasil
33	4240	-	Loss
34	4255	-	Loss
Presentase Keberhasilan			88.3%

Tabel 4.12 Pengujian Komunikasi Data LED 2 Warna Hijau

No	Data yang Dikirim	Data yang Diterima	Status
1	5000	5000	Berhasil
2	5005	5005	Berhasil
3	5010	-	Loss
4	5015	5015	Berhasil
5	5020	5020	Berhasil
6	5025	5025	Berhasil
7	5030	5030	Berhasil
8	5035	5035	Berhasil
9	5040	5040	Berhasil
10	5045	5045	Berhasil
11	5050	5050	Berhasil
12	5055	5055	Berhasil
13	5060	5060	Berhasil
14	5065	5065	Berhasil
15	5070	5070	Berhasil
16	5075	5075	Berhasil
17	5080	5080	Berhasil
18	5090	5090	Berhasil
19	5100	5100	Berhasil
20	5110	5110	Berhasil
21	5120	5120	Berhasil
22	5130	5130	Berhasil
23	5140	5140	Berhasil
24	5150	5150	Berhasil
25	5160	5160	Berhasil
26	5170	5170	Berhasil
27	5180	5180	Berhasil
28	5190	5190	Berhasil
29	5200	5200	Berhasil
30	5210	5210	Berhasil
31	5220	5220	Berhasil
32	5230	5230	Berhasil
33	5240	5240	Berhasil
34	5255	5255	Berhasil
Presentase Keberhasilan			97.1%

Tabel 4.13 Pengujian Komunikasi Data LED 2 Warna Biru

No	Data yang Dikirim	Data yang Diterima	Status
1	6000	6000	Berhasil
2	6005	6005	Berhasil
3	6010	6010	Berhasil
4	6015	6015	Berhasil
5	6020	6020	Berhasil
6	6025	6025	Berhasil
7	6030	6030	Berhasil
8	6035	6035	Berhasil
9	6040	6040	Berhasil
10	6045	6045	Berhasil
11	6050	6050	Berhasil
12	6055	6055	Berhasil
13	6060	6060	Berhasil
14	6065	6065	Berhasil
15	6070	6070	Berhasil
16	6075	6075	Berhasil
17	6080	6080	Berhasil
18	6090	6090	Berhasil
19	6100	6100	Berhasil
20	6110	6110	Berhasil
21	6120	6120	Berhasil
22	6130	6130	Berhasil
23	6140	6140	Berhasil
24	6150	6150	Berhasil
25	6160	6160	Berhasil
26	6170	6170	Berhasil
27	6180	6180	Berhasil
28	6190	6190	Berhasil
29	6200	6200	Berhasil
30	6210	6210	Berhasil
31	6220	6220	Berhasil
32	6230	6230	Berhasil
33	6240	6240	Berhasil
34	6255	-	Loss
Presentasi Keberhasilan			97.1%

Tabel 4.14 Pengujian Komunikasi Data LED 3 Warna Merah

No	Data yang Dikirim	Data yang Diterima	Status
1	7000	7000	Berhasil
2	7005	7005	Berhasil
3	7010	7010	Berhasil
4	7015	7015	Berhasil
5	7020	7020	Berhasil
6	7025	7025	Berhasil
7	7030	7030	Berhasil
8	7035	7035	Berhasil
9	7040	7040	Berhasil
10	7045	7045	Berhasil
11	7050	7050	Berhasil
12	7055	-	Loss
13	7060	7060	Berhasil
14	7065	7065	Berhasil
15	7070	7070	Berhasil
16	7075	7075	Berhasil
17	7080	7080	Berhasil
18	7090	7090	Berhasil
19	7100	7100	Berhasil
20	7110	7110	Berhasil
21	7120	7120	Berhasil
22	7130	7130	Berhasil
23	7140	7140	Berhasil
24	7150	7150	Berhasil
25	7160	7160	Berhasil
26	7170	-	Loss
27	7180	7180	Berhasil
28	7190	7190	Berhasil
29	7200	7200	Berhasil
30	7210	7210	Berhasil
31	7220	7220	Berhasil
32	7230	7230	Berhasil
33	7240	7240	Berhasil
34	7255	7255	Berhasil
Presentase Keberhasilan			94.2%

Tabel 4.15 Pengujian Komunikasi Data LED 3 Warna Hijau

No	Data yang Dikirim	Data yang Diterima	Status
1	8000	8000	Berhasil
2	8005	8005	Berhasil
3	8010	8010	Berhasil
4	8015	8015	Berhasil
5	8020	8020	Berhasil
6	8025	8025	Berhasil
7	8030	8030	Berhasil
8	8035	8035	Berhasil
9	8040	8040	Berhasil
10	8045	8045	Berhasil
11	8050	8050	Berhasil
12	8055	8055	Berhasil
13	8060	8060	Berhasil
14	8065	8065	Berhasil
15	8070	8070	Berhasil
16	8075	8075	Berhasil
17	8080	8080	Berhasil
18	8090	8090	Berhasil
19	8100	8100	Berhasil
20	8110	8110	Berhasil
21	8120	8120	Berhasil
22	8130	8130	Berhasil
23	8140	8140	Berhasil
24	8150	8150	Berhasil
25	8160	8160	Berhasil
26	8170	8170	Berhasil
27	8180	8180	Berhasil
28	8190	8190	Berhasil
29	8200	8200	Berhasil
30	8210	8210	Berhasil
31	8220	8220	Berhasil
32	8230	8230	Berhasil
33	8240	8240	Berhasil
34	8255	8255	Berhasil
Presentase Keberhasilan			100%

Tabel 4.16 Pengujian Komunikasi Data LED 3 Warna Biru

No	Data yang Dikirim	Data yang Diterima	Status
1	9000	9000	Berhasil
2	9005	9005	Berhasil
3	9010	9010	Berhasil
4	9015	9015	Berhasil
5	9020	9020	Berhasil
6	9025	9025	Berhasil
7	9030	-	Loss
8	9035	9035	Berhasil
9	9040	9040	Berhasil
10	9045	9045	Berhasil
11	9050	9050	Berhasil
12	9055	-	Loss
13	9060	9060	Berhasil
14	9065	9065	Berhasil
15	9070	9070	Berhasil
16	9075	-	Loss
17	9080	9080	Berhasil
18	9090	9090	Berhasil
19	9100	9100	Berhasil
20	9110	9110	Berhasil
21	9120	9120	Berhasil
22	9130	9130	Berhasil
23	9140	9140	Berhasil
24	9150	9150	Berhasil
25	9160	9160	Berhasil
26	9170	9170	Berhasil
27	9180	9180	Berhasil
28	9190	9190	Berhasil
29	9200	9200	Berhasil
30	9210	9210	Berhasil
31	9220	9220	Berhasil
32	9230	9230	Berhasil
33	9240	9240	Berhasil
34	9255	9255	Berhasil
Presentase Keberhasilan			91.2%

Tabel 4.17 Pengujian komunikasi Raspberry pi dengan Arduino

No	Data yang dikirim (R)	Data yang dikirim (G)	Data yang dikirim (B)	Data yang diterima (R)	Data yang diterima (G)	Data yang diterima (B)
1	62	48	36	62	48	36
2	63	50	40	63	50	40
3	64	51	42	64	51	42
4	66	53	44	66	53	44
5	67	57	46	67	57	46
6	69	59	49	69	59	49
7	71	64	51	71	64	51
8	75	65	53	75	65	53
9	77	69	55	77	69	55
10	79	70	57	79	70	57
11	82	72	60	82	72	60
12	88	74	62	88	74	62
13	90	77	66	90	77	66
14	91	85	69	91	85	69
15	93	89	72	93	89	72
16	94	91	74	94	91	74
17	96	97	83	96	97	83
18	98	100	90	98	100	90
19	99	115	97	99	115	97
20	100	120	102	100	120	102
21	102	126	107	102	126	107
22	103	128	111	103	128	111
23	112	130	119	112	130	119
24	114	131	120	114	131	120
25	115	133	123	115	133	123
26	117	135	126	117	135	126
27	118	138	128	118	138	128
28	119	140	129	119	140	129
29	120	143	130	120	143	130
30	122	147	131	122	147	131
31	123	148	134	123	148	134
Presentase Keberhasilan						100%

Tabel 4.18 Pengujian program animasi *auto-fading* periode 5 detik

Waktu (ms)	Merah	Hijau	Biru
10	0	14	195
100	0	0	205
200	0	0	229
300	0	0	247
400	0	0	254
500	0	0	253
600	0	0	243
700	3	0	226
800	45	0	203
900	87	0	174
1000	128	0	139
1500	251	0	0
2000	189	67	0
2500	0	231	0
3000	0	224	0
3500	0	54	167
4000	0	0	254
4500	115	0	151
5000	248	0	0

Tabel 4.19 Pengujian program animasi *auto-fading* periode 8 detik

Waktu (ms)	Merah	Hijau	Biru
10	0	68	155
100	0	57	164
250	0	17	193
500	0	0	230
1000	0	0	251
1500	59	0	194
2000	180	0	79
2500	247	0	0
3000	240	0	0
3500	164	100	0
4000	37	208	0
4500	0	254	0
5000	0	219	0
5500	0	122	105
6000	0	0	209
6500	0	0	254
7000	19	0	218
7500	148	0	118
8000	233	0	0

Tabel 4.20 Pengujian program animasi *auto-fading* periode 10 detik

Waktu (ms)	Merah	Hijau	Biru
1000	0	204	0
1500	0	124	103
2000	0	19	191
2500	0	0	245
3000	0	0	249
3500	44	0	204
4000	147	0	120
4500	220	0	14
5000	254	0	0
5500	235	0	0
6000	175	87	0
6500	78	182	0
7000	0	239	0
7500	0	253	0
8000	0	234	0
8500	0	32	182
9000	0	32	182
9500	0	0	241
10000	0	0	252

Tabel 4.21 Pengujian program *music-synched*

Waktu (ms)	Merah	Hijau	Biru
80	0	5	200
180	0	0	225
280	0	0	243
350	0	0	251
470	0	0	254
600	0	0	243
690	0	0	228
720	11	0	222
830	58	0	195
900	87	0	174
920	95	0	168
940	103	0	161
990	124	0	143
1020	136	0	131
1060	151	0	114
1100	166	0	98
1140	179	0	81
1200	196	0	56
1320	225	0	5

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

PROGRAM KODE BLOK APP INVENTOR



```

when animation.Click
do
  set MenuScreen.Visible to false
  set AnimationScreen.Visible to true

```

```

when Animation_1.Click
do
  if Animation_1.Text = "Animation 1 ON"
  then
    set Animation_1.Text to "Animation 1 OFF"
    call BluetoothClient1.SendByteNumber
      number 7
  else
    set Animation_1.Text to "Animation 1 ON"
    call BluetoothClient1.SendByteNumber
      number 6
  end if

```

```

when Connect.AfterPicking
do
  if
  then
    call BluetoothClient1.Connect
      address Connect.Selection
    set led1.Enabled to true
    set led2.Enabled to true
    set led3.Enabled to true
    set animation.Enabled to true
    set Connect.Enabled to false
    set Disconnect.Enabled to true
    set Connecting.Text to "CONNECTED"
    set Connecting.TextColor to
    set Ball1.PaintColor to
  end if

```

```

when Connect.BeforePicking
do
  set Connect.Elements to BluetoothClient1.AddressesAndNames

```

```

when Disconnect .Click
do
  call BluetoothClient1 .Disconnect
  set led1 . Enabled to false
  set led2 . Enabled to false
  set led3 . Enabled to false
  set animation . Enabled to false
  set Connect . Enabled to true
  set Disconnect . Enabled to false
  set Connecting . Text to "NOT CONNECTED"
  set Connecting . TextColor to red
  set Ball1 . PaintColor to red

```

```

when led1 .Click
do
  set MenuScreen . Visible to false
  set Led1Screen . Visible to true

```

```

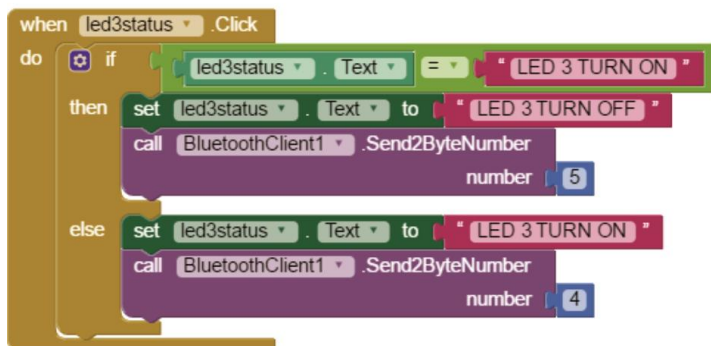
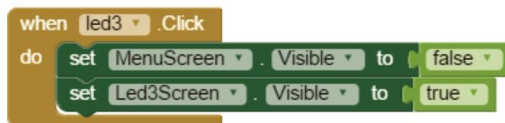
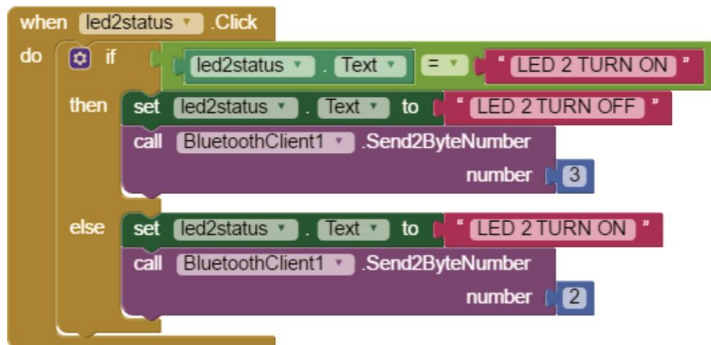
when led1status .Click
do
  if led1status . Text = "LED 1 TURN ON"
  then
    set led1status . Text to "LED 1 TURN OFF"
    call BluetoothClient1 .Send2ByteNumber
      number 1
  else
    set led1status . Text to "LED 1 TURN ON"
    call BluetoothClient1 .Send2ByteNumber
      number 0
  end if

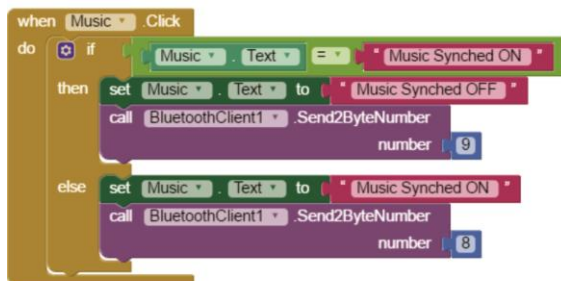
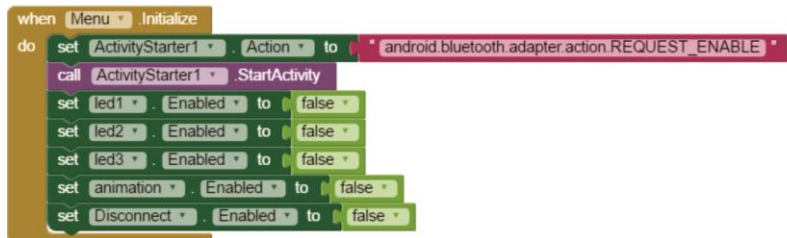
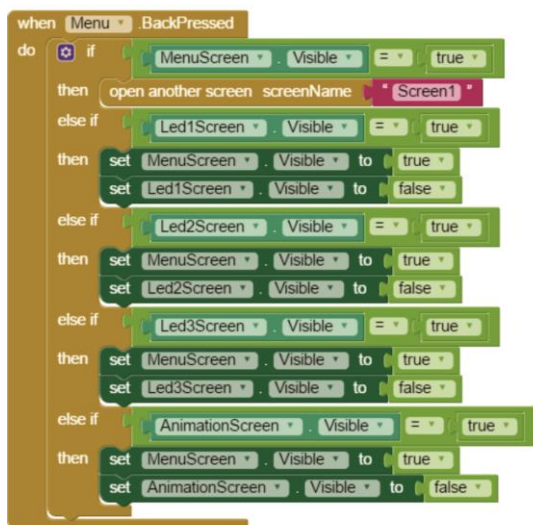
```

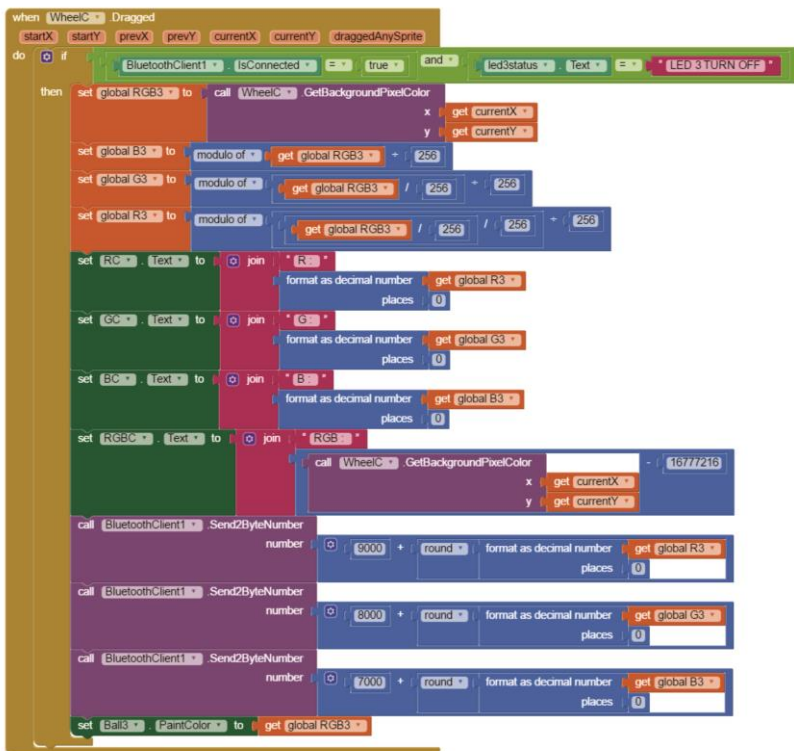
```

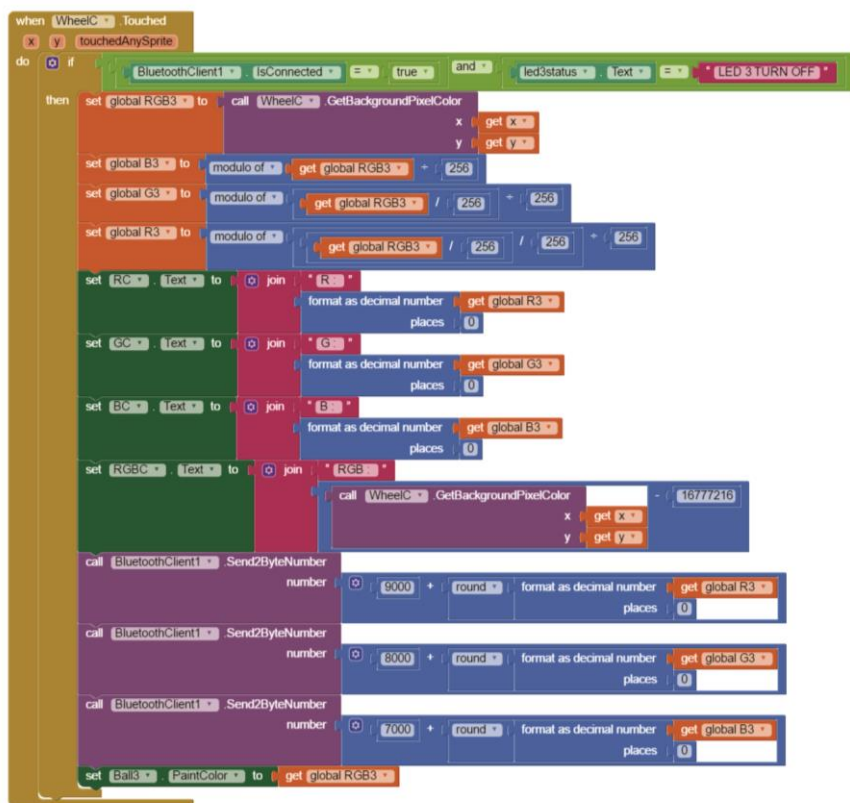
when led2 .Click
do
  set MenuScreen . Visible to false
  set Led2Screen . Visible to true

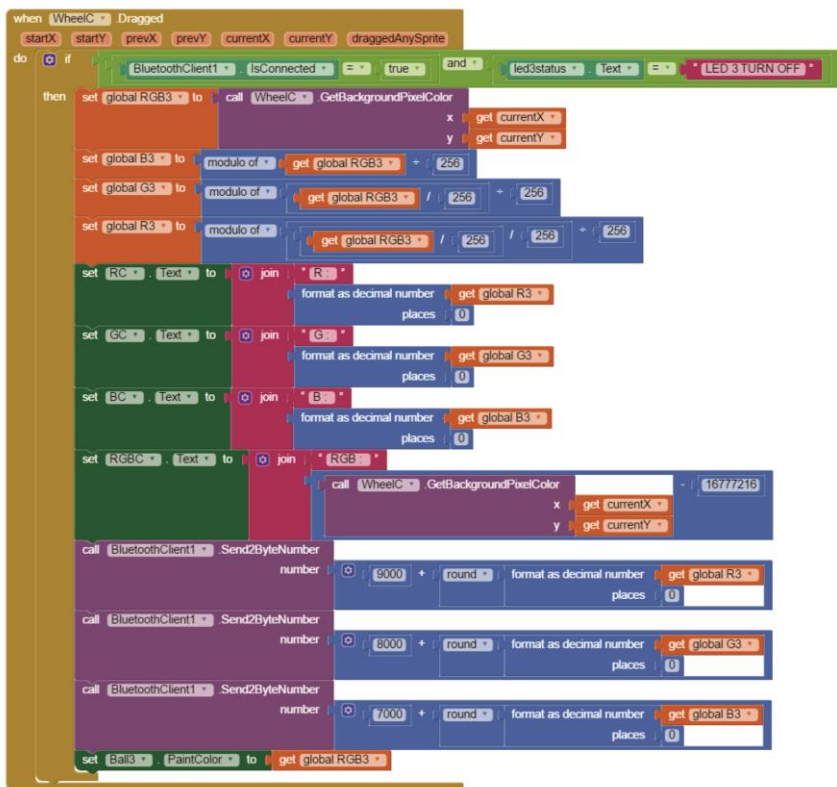
```

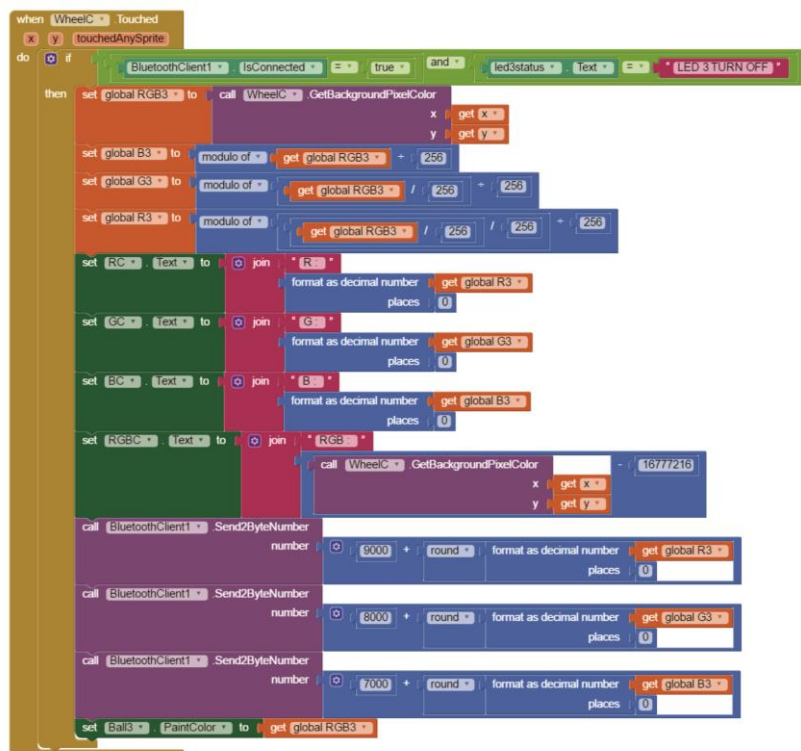


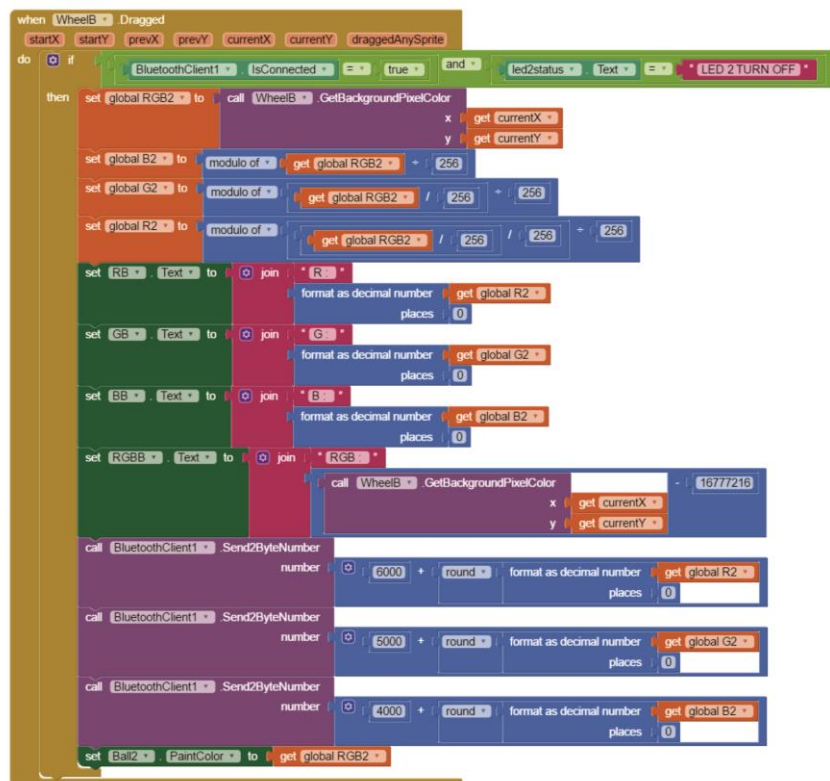


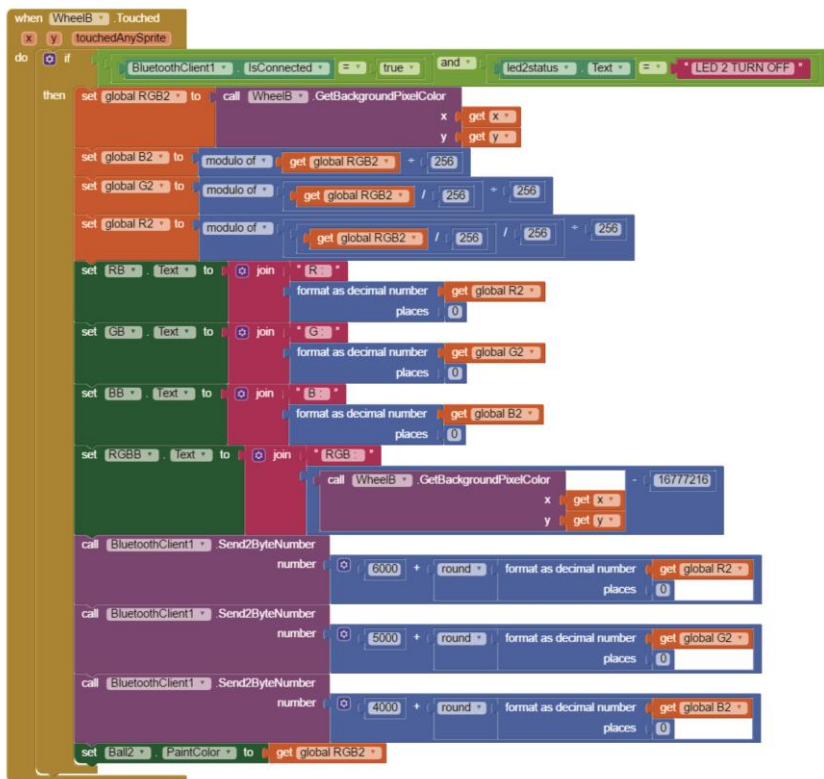












-----Halaman ini sengaja dikosongkan-----

LAMPIRAN C

PROGRAM ARDUINO

```
#define bluetooth Serial3
#define delayTime 20
#define musicSensor A8
#define LDRSensor A0
```

```
double Setpoint;
double OutLDR;
long time = 0;
int musicVal = 0;
int redVal1 = 0;
int greenVal1 = 0;
int blueVal1 = 0;
int redVal2 = 0;
int greenVal2 = 0;
int blueVal2 = 0;
int redVal3 = 0;
int greenVal3 = 0;
int blueVal3 = 0;
int rMax1 = 0;
int gMax1 = 0;
int bMax1 = 0;
int rMax2 = 0;
int gMax2 = 0;
int bMax2 = 0;
int rMax3 = 0;
int gMax3 = 0;
int bMax3 = 0;
int periode = 0;
int redMusic = 0;
int greenMusic = 0;
int blueMusic = 0;
unsigned int color;
unsigned int color1;
unsigned int color2;
String dataIn;
String dt[10];
```

```

int i;
boolean parsing = false;
int Rx, Gx, Bx;

void setup()
{
  bluetooth.begin(9600);
  Serial.begin(9600);
  while (!Serial) {
    ;
  }
  establishContact();
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(musicSensor, INPUT);
  dataIn = "";

  TCCR1B = (TCCR1B & 0xF8) | 0x03 ;
  TCCR2B = (TCCR2B & 0xF8) | 0x04 ;
  TCCR3B = (TCCR3B & 0xF8) | 0x04 ;
  TCCR4B = (TCCR4B & 0xF8) | 0x04 ;

  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
  digitalWrite(11, LOW);

```



```

}

void loop()
{

  if (Serial.available() > 0) {
    char inChar = (char)Serial.read();
    dataIn += inChar;
    if (inChar == '\n') {
      parsing = true;
    }

    if (parsing) {
      parsingData();
      parsing = false;
      dataIn = "";
    }

    analogWrite(2, Rx);
    analogWrite(3, Rx);
    analogWrite(5, Rx);
    analogWrite(6, Gx);
    analogWrite(7, Gx);
    analogWrite(8, Gx);
    analogWrite(9, Bx);
    analogWrite(10, Bx);
    analogWrite(11, Bx);
    delay(1);
  }

  if (bluetooth.available() >= 2 )
  {
    //Serial.println(color);
    color1 = bluetooth.read();
    color2 = bluetooth.read();
    color = (color2 * 256) + color1;

    if (color == 1) {
      digitalWrite(2, HIGH);
    }
  }
}

```

```

digitalWrite(6, HIGH);
digitalWrite(9, HIGH);
}

else if (color >= 1000 && color <= 1255) {
    int blueA = color;
    blueA = map(blueA, 1000, 1255, 0, 255);
    int blueA1 = blueA;
    analogWrite(9, blueA);
    delay( delayTime );
}

else if (color >= 2000 && color <= 2255) {
    int greenA = color;
    greenA = map(greenA, 2000, 2255, 0, 255);
    int greenA1 = greenA;
    analogWrite(6, greenA1);
    delay( delayTime );
}

else if (color >= 3000 && color <= 3255) {
    int redA = color;
    redA = map(redA, 3000, 3255, 0, 255);
    int redA1 = redA;
    analogWrite(2, redA1);
    delay( delayTime );
}

else if (color == 0) {
    digitalWrite(2, LOW);
    digitalWrite(6, LOW);
    digitalWrite(9, LOW);
}

else if (color == 3) {
    digitalWrite(3, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(10, HIGH);
}

```

```

else if (color >= 4000 && color <= 4256) {
    int blueB = color;
    blueB = map(blueB, 4000, 4256, 0, 255);
    int blueB1 = blueB;
    analogWrite(10, blueB1);
    delay( delayTime );
}

else if (color >= 5000 && color <= 5256) {
    int greenB = color;
    greenB = map(greenB, 5000, 5256, 0, 255);
    int greenB1 = greenB;
    analogWrite(7, greenB1);
    delay( delayTime);
}

else if (color >= 6000 && color <= 6256) {
    int redB = color;
    redB = map(redB, 6000, 6256, 0, 255);
    int redB1 = redB;
    analogWrite(3, redB1);
    delay( delayTime );
}

else if (color == 2) {
    digitalWrite(3, LOW);
    digitalWrite(7, LOW);
    digitalWrite(10, LOW);
}

else if (color == 5) {
    digitalWrite(5, HIGH);
    digitalWrite(8, HIGH);
    digitalWrite(11, HIGH);
}

else if (color >= 7000 && color <= 7256) {
    int blueC = color;

```

```

blueC = map(blueC, 7000, 7256, 0, 255);
int blueC1 = blueC;
analogWrite(11, blueC1);
delay( delayTime );
}

else if (color >= 8000 && color <= 8256) {
    int greenC = color;
    greenC = map(greenC, 8000, 8256, 0, 255);
    int greenC1 = greenC;
    analogWrite(8, greenC1);
    delay ( delayTime );
}

else if (color >= 9000 && color < 9256) {
    int redC = color;
    redC = map(redC, 9000, 9256, 0, 255);
    int redC1 = redC;
    analogWrite(5, redC1);
    delay( delayTime );
}

else if (color == 4) {
    digitalWrite(11, LOW);
    digitalWrite(8, LOW);
    digitalWrite(5, LOW);
}
else if (color >= 10000 && color <= 10255) {
    Setpoint = color;
    Setpoint = map(Setpoint, 10000, 10255, 0, 255);
}
else if (color >= 10500 && color <= 10510) {
    periode = color;
    periode = map(periode, 10500, 10510, 0, 10000);
}

else if (color >= 11000 && color <= 11100) {
    rMax1 = color;

```

```

    rMax1 = map(rMax1, 11000, 11100, 0, 2000);
}

else if (color >= 11500 && color <= 11600) {
    gMax1 = color;
    gMax1 = map(gMax1, 11500, 11600, 0, 2000);
}

else if (color >= 12000 && color <= 12100) {
    bMax1 = color;
    bMax1 = map(bMax1, 12000, 12100, 0, 2000);
}

else if (color >= 12500 && color <= 12600) {
    rMax2 = color;
    rMax2 = map(rMax2, 12500, 12600, 0, 2000);
}

else if (color >= 13000 && color <= 13100) {
    gMax2 = color;
    gMax2 = map(gMax2, 13000, 13100, 0, 2000);
}

else if (color >= 13500 && color <= 13600) {
    bMax2 = color;
    bMax2 = map(bMax2, 13500, 13600, 0, 2000);
}

else if (color >= 14000 && color <= 14100) {
    rMax3 = color;
    rMax3 = map(rMax3, 14000, 14100, 0, 2000);
}

else if (color >= 14500 && color <= 14600) {
    gMax3 = color;
    gMax3 = map(gMax3, 14500, 14600, 0, 2000);
}

else if (color >= 15000 && color <= 15100) {

```

```

    bMax3 = color;
    bMax3 = map(bMax3, 15000, 15100, 0, 2000);
  }
}
if (color == 7 ) {
  musicVal = analogRead(musicSensor);
  musicVal = map(musicVal, 0, 1023, 0, 255);
  musicShow();
  if (musicVal > 130) {
    if (redMusic > 0) {
      analogWrite(2, redMusic);
      analogWrite(3, redMusic);
      analogWrite(5, redMusic);
      delay(30);
    }
    else
    {
      analogWrite(2, 0);
      analogWrite(3, 0);
      analogWrite(5, 0);
    }
    if (greenMusic > 0) {
      analogWrite(6, greenMusic);
      analogWrite(7, greenMusic);
      analogWrite(8, greenMusic);
      delay(30);
    }
    else
    {
      analogWrite(6, 0);
      analogWrite(7, 0);
      analogWrite(8, 0);
    }
    if (blueMusic > 0) {
      analogWrite(9, blueMusic);
      analogWrite(10, blueMusic);
      analogWrite(11, blueMusic);
      delay(30);
    }
  }
}

```

```

else
{
    analogWrite(9, 0);
    analogWrite(10, 0);
    analogWrite(11, 0);
}
}
else {
    analogWrite(2, 0);
    analogWrite(3, 0);
    analogWrite(5, 0);
    analogWrite(6, 0);
    analogWrite(7, 0);
    analogWrite(8, 0);
    analogWrite(9, 0);
    analogWrite(10, 0);
    analogWrite(11, 0);
}
}
else if (color == 13) {
    rgbFading1();
    rgbWrite1();
    rgbFading2();
    rgbWrite2();
    rgbFading3();
    rgbWrite3();
    Serial.print(redVal1);
    Serial.print(";");
    Serial.print(greenVal1);
    Serial.print(";");
    Serial.println(blueVal1);
}
else if (color == 14) {
    double LDR = analogRead(LDRSensor);
    LDR = map(LDR, 1023, 0, 0, 255);
    double error = Setpoint - LDR;

    double Proportional = 1.12 * error;
    OutLDR = LDR + Proportional;

```

```

    if (OutLDR < 0) {
        OutLDR = 0;
    }
    else if (OutLDR > 255) {
        OutLDR = 255;
    }
    //myPID.Compute();
    analogWrite(2, OutLDR);
    analogWrite(3, OutLDR);
    analogWrite(5, OutLDR);
    analogWrite(6, OutLDR);
    analogWrite(7, OutLDR);
    analogWrite(8, OutLDR);
    analogWrite(9, OutLDR);
    analogWrite(10, OutLDR);
    analogWrite(11, OutLDR);
    //Serial.print(Setpoint);
    //Serial.print(";");
    //Serial.print(OutLDR);
    //Serial.print(";");
    //Serial.println(LDR);
    delay(0.1);
}
else if (color == 6 || color == 8 || color == 10) {
    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(5, LOW);
    digitalWrite(6, LOW);
    digitalWrite(7, LOW);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, LOW);
}
}
void parsingData() {
    int j = 0;

    Serial.print("data masuk : ");

```



```

Serial.print(dataIn);
Serial.print("\n");
dt[j] = "";
for (i = 1; i < dataIn.length(); i++) {
  if ((dataIn[i] == '#') || (dataIn[i] == ',')) {
    j++;
    dt[j] = "";
  }
  else {
    dt[j] = dt[j] + dataIn[i];
  }
}
Rx = dt[0].toInt();
Gx = dt[1].toInt();
Bx = dt[2].toInt();
// kirim data hasil parsing
Serial.print("data R : ");
Serial.print(Rx);
Serial.print("\n");
Serial.print("data G : ");
Serial.print(Gx);
Serial.print("\n");
Serial.print("data B : ");
Serial.print(Bx);
Serial.print("\n\n");
}

void musicShow() {
  time = millis();
  int val1 = 255 * sin(((2 * PI / 4000) * time));
  int val2 = 255 * sin(((2 * PI / 4000) * time) + 90);
  int val3 = 255 * sin(((2 * PI / 4000) * time) - 90);
  redMusic = val1;
  greenMusic = val2;
  blueMusic = val3;
}

void rgbWrite1() {
  if (redVal1 > 0) {

```

```

    analogWrite(2, redVal1);
}
else
{
    analogWrite(2, 0);
}
if (greenVal1 > 0) {
    analogWrite(6, greenVal1);
}
else
{
    analogWrite(6, 0);
}
if (blueVal1 > 0) {
    analogWrite(9, blueVal1);
}
else
{
    analogWrite(9, 0);
}
}
void rgbWrite2() {
    if (redVal2 > 0) {
        analogWrite(3, redVal2);
    }
    else
    {
        analogWrite(3, 0);
    }
    if (greenVal2 > 0) {
        analogWrite(7, greenVal2);
    }
    else
    {
        analogWrite(7, 0);
    }
    if (blueVal2 > 0) {
        analogWrite(10, blueVal2);
    }
}

```

```

else
{
    analogWrite(10, 0);
}
}
void rgbWrite3() {

    if (redVal3 > 0) {
        analogWrite(5, redVal3);
    }
    else
    {
        analogWrite(5, 0);
    }
    if (greenVal3 > 0) {
        analogWrite(8, greenVal3);
    }
    else
    {
        analogWrite(8, 0);
    }
    if (blueVal3 > 0) {
        analogWrite(11, blueVal3);
    }
    else
    {
        analogWrite(11, 0);
    }
}

void rgbFading1() {
    time = millis();
    int value1 = 255 * sin((2 * PI / periode) * (rMax1 - time));
    int value2 = 255 * sin((2 * PI / periode) * (gMax1 - time) + 90);
    int value3 = 255 * sin((2 * PI / periode) * (bMax1 - time) - 90);
    redVal1 = value1;
    greenVal1 = value2;
    blueVal1 = value3;
}

```

```

}

void rgbFading2() {
    time = millis();
    int value4 = 255 * sin((2 * PI / periode) * (rMax2 - time));
    int value5 = 255 * sin((2 * PI / periode) * (gMax2 - time) + 90);
    int value6 = 255 * sin((2 * PI / periode) * (bMax2 - time) - 90);
    redVal2 = value4;
    greenVal2 = value5;
    blueVal2 = value6;
}

void rgbFading3() {
    time = millis();
    int value7 = 255 * sin((2 * PI / periode) * (rMax3 - time));
    int value8 = 255 * sin((2 * PI / periode) * (gMax3 - time) + 90);
    int value9 = 255 * sin((2 * PI / periode) * (bMax3 - time) - 90);
    redVal3 = value7;
    greenVal3 = value8;
    blueVal3 = value9;
}

void establishContact() {
    while (Serial.available() <= 0) {
        Serial.println("Ready to Start");
        delay(300);
    }
}

```

LAMPIRAN D

PROGRAM RASPBERRY PI (PYTHON)

```
import numpy
import cv2
import serial
import math

ser = serial.Serial('/dev/ttyACM0',115200,timeout=1)
cam = cv2.VideoCapture(-1)
cam.set(3, 320)
cam.set(4, 480)

while(True):
    _, frame = cam.read()
    ROI = frame [220:240,300:320]
    nilai = ROI[10,10]
    b = nilai[0]
    g = nilai[1]
    r = nilai [2]
    print nilai
    print ('r : %s' %r)
    print ('g : %s' %g)
    print ('b : %s' %b)

    ser.write('*%3d,' %r)
    ser.write('%3d,' %g)
    ser.write('%3d#\n' %b)
    cv2.rectangle(frame,(300,220),(320,240),(0,255,0),3)
    cv2.imshow('sensor',frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN E DATASHEET

1. Datasheet Arduino Mega

MEGA 2560 R3 Datasheet



■ INTRODUCTION

MEGA 2560 R3 is an Arduino compatible board, It's based on Arduino MEGA 2560 design, So you can use BUONO MEGA 2560 as Arduino MEGA 2560, All Code, shield and IDE for Arduino MEGA 2560 R3 are also valid on BUONO MEGA 2560. Some visible improvement on hardware make BUONO more flexible and easier use. For example: 3.3V or 5V IO selectable allow you connect some 3.3V modules (such as XBee) to BUONO directly.



■ THE DIFFERENCE BETWEEN Arduino MEGA 2560 AND BUONO MEGA 2560

	Arduino MEGA 2560	BUONO MEGA 2560
USB Connector	Type B Female	Micro USB connector
Operating Voltage	5V	3.3V or 5V selectable
3.3V Current	50mA	600mA
5V Current	500mA	1.8A
Input Range	7V to 12V	6V to 23V
MCU	ATMEGA2560	ATMEGA2560
Others		Ground terminal LPPF for AVCC

■ FEATURES

- Inherits all of Arduino MEGA 2560's features
- Compatible to Arduino MEGA 2560's pin layout, screw hole and dimensions
- 3.3V or 5V operating voltage selectable
- More visible location of indication LEDs
- Evolved with SMD components
- Digital I/O pins 54 (of which 15 provided PWM output)
- Analog Input pins 16
- Flash memory 256 KB or which 8 KB used by bootloader
- SRAM 8 KB
- EEPROM 4 KB
- Clock speed 16 MHz
- Micro USB connection make the cable can be mixed use with most digital products such as cell phone
- 5V DC/DC provide 95% efficiency
- 6V to 23V wide range input
- Provide max 1.8A for peripheral circuits
- Ground terminal make measurement easier

■ PACKAGE LIST

- 1x BUONO MEGA 2560
- 1x Micro USB cable
- 4x Screw and nut

■ CONTACT US

540 Mill River LN San Jose,
CA 95134
E-mail: support@inhaos.com
Web: www.inhaos.com

How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board select**

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.



Done compiling

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

2. Datasheet IRF 540



IRF540, SiHF540

Vishay Siliconix

Power MOSFET

PRODUCT SUMMARY		
V_{DS} (V)		100
$R_{DS(on)}$ (Ω)	$V_{GS} = 10\text{ V}$	0.077
Q_g (Max.) (nC)		72
Q_{gs} (nC)		11
Q_{gd} (nC)		32
Configuration		Single

FEATURES

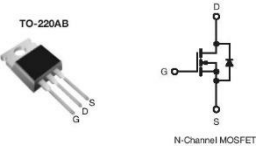
- Dynamic dV/dt Rating
- Repetitive Avalanche Rated
- 175 °C Operating Temperature
- Fast Switching
- Ease of Paralleling
- Simple Drive Requirements
- Compliant to RoHS Directive 2002/95/EC



RoHS*
COMPLIANT

DESCRIPTION

Third generation Power MOSFETs from Vishay provide the designer with the best combination of fast switching, ruggedized device design, low on-resistance and cost-effectiveness. The TO-220AB package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 W. The low thermal resistance and low package cost of the TO-220AB contribute to its wide acceptance throughout the industry.



ORDERING INFORMATION	
Package	TO-220AB
Lead (Pb)-free	IRF540PbF
	SiHF540-E3
SnPb	IRF540
	SiHF540

ABSOLUTE MAXIMUM RATINGS ($T_C = 25\text{ °C}$, unless otherwise noted)				
PARAMETER		SYMBOL	LIMIT	UNIT
Drain-Source Voltage		V_{DS}	100	V
Gate-Source Voltage		V_{GS}	± 20	
Continuous Drain Current	V_{GS} at 10 V	$T_C = 25\text{ °C}$	28	A
			20	
		$T_C = 100\text{ °C}$	10	
Pulsed Drain Current ^a		I_{DM}	110	
Linear Derating Factor			1.0	W/°C
Single Pulse Avalanche Energy ^b		E_{AS}	230	mJ
Repetitive Avalanche Current ^c		I_{AR}	28	A
Repetitive Avalanche Energy ^d		E_{AR}	15	mJ
Maximum Power Dissipation	$T_C = 25\text{ °C}$	P_D	150	W
Peak Diode Recovery dV/dt ^c		dV/dt	5.5	V/ns
Operating Junction and Storage Temperature Range		T_J, T_{stg}	-55 to +175	°C
Soldering Recommendations (Peak Temperature)	for 10 s		300 ^d	
Mounting Torque	6-32 or M3 screw		10	lbf · in
			1.1	N · m

Notes

- Repetitive rating; pulse width limited by maximum junction temperature (see fig. 11).
- $V_{DS} = 25\text{ V}$, starting $T_J = 25\text{ °C}$, $L = 440\text{ }\mu\text{H}$, $R_{\theta JA} = 25\text{ }\Omega$, $I_{AS} = 28\text{ A}$ (see fig. 12).
- $I_{AS} \leq 28\text{ A}$, $dI/dt \leq 170\text{ A}/\mu\text{s}$, $V_{DS} \leq V_{GS}$, $T_J \leq 175\text{ °C}$.
- 1.6 mm from case.

* Pb containing terminations are not RoHS compliant, exemptions may apply

Document Number: 91021
S11-0510-Rev. B, 21-Mar-11

www.vishay.com

This datasheet is subject to change without notice.



THE PRODUCT DESCRIBED HEREIN AND THIS DATASHEET ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc?91000

IRF540, SiHF540

Vishay Siliconix

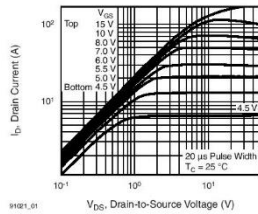
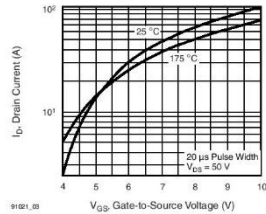
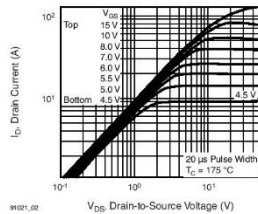
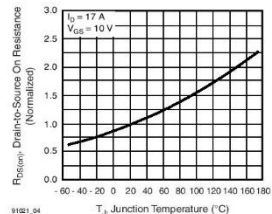


THERMAL RESISTANCE RATINGS				
PARAMETER	SYMBOL	TYP.	MAX.	UNIT
Maximum Junction-to-Ambient	$R_{\theta JA}$	-	62	°C/W
Case-to-Sink, Flat, Greased Surface	$R_{\theta CS}$	0.50	-	
Maximum Junction-to-Case (Drain)	$R_{\theta JC}$	-	1.0	

SPECIFICATIONS (T _J = 25 °C, unless otherwise noted)						
PARAMETER	SYMBOL	TEST CONDITIONS	MIN.	TYP.	MAX.	UNIT
Static						
Drain-Source Breakdown Voltage	V _{DS}	V _{GS} = 0 V, I _D = 250 μA	100	-	-	V
V _{DS} Temperature Coefficient	ΔV _{DS} /T _J	Reference to 25 °C, I _D = 1 mA	-	0.13	-	V/°C
Gate-Source Threshold Voltage	V _{GS(th)}	V _{DS} = V _{GS} , I _D = 250 μA	2.0	-	4.0	V
Gate-Source Leakage	I _{GSS}	V _{GS} = ± 20 V	-	-	± 100	nA
Zero Gate Voltage Drain Current	I _{DSS}	V _{DS} = 100 V, V _{GS} = 0 V	-	-	25	μA
		V _{DS} = 80 V, V _{GS} = 0 V, T _J = 150 °C	-	-	250	
Drain-Source On-State Resistance	R _{DS(on)}	V _{GS} = 10 V, I _D = 17 A ^b	-	-	0.077	Ω
Forward Transconductance	g _{fs}	V _{DS} = 50 V, I _D = 17 A ^b	8.7	-	-	S
Dynamic						
Input Capacitance	C _{iss}	V _{GS} = 0 V, V _{DS} = 25 V, f = 1.0 MHz, see fig. 5	-	1700	-	pF
Output Capacitance	C _{oss}		-	560	-	
Reverse Transfer Capacitance	C _{rss}		-	120	-	
Total Gate Charge	Q _g	V _{GS} = 10 V, I _D = 17 A, V _{DS} = 80 V, see fig. 6 and 13 ^b	-	-	72	nC
Gate-Source Charge	Q _{gs}		-	-	11	
Gate-Drain Charge	Q _{gd}		-	-	32	
Turn-On Delay Time	t _{d(on)}	V _{DS} = 50 V, I _D = 17 A R _{θJA} = 9.1 °C/W, R _{θJB} = 2.9 °C/W, see fig. 10 ^b	-	11	-	ns
Rise Time	t _r		-	44	-	
Turn-Off Delay Time	t _{d(off)}		-	53	-	
Fall Time	t _f		-	43	-	
Internal Drain Inductance	L _D	Between lead, 6 mm (0.25") from package and center of die contact 	-	4.5	-	nH
Internal Source Inductance	L _S		-	7.5	-	
Drain-Source Body Diode Characteristics						
Continuous Source-Drain Diode Current	I _S	MOSFET symbol showing the integral reverse p-n junction diode 	-	-	28	A
Pulsed Diode Forward Current ^a	I _{SM}		-	-	110	
Body Diode Voltage	V _{SD}	T _J = 25 °C, I _S = 28 A, V _{GS} = 0 V ^b	-	-	2.5	V
Body Diode Reverse Recovery Time	t _{rr}	T _J = 25 °C, I _S = 17 A, di/dt = 100 A/μs ^b	-	180	360	ns
Body Diode Reverse Recovery Charge	Q _{rr}		-	1.3	2.8	μC
Forward Turn-On Time	t _{on}	Intrinsic turn-on time is negligible (turn-on is dominated by L _S and L _D)				

Notes

- a. Repetitive rating; pulse width limited by maximum junction temperature (see fig. 11).
b. Pulse width $\leq 300\text{ }\mu\text{s}$; duty cycle $\leq 2\%$.

TYPICAL CHARACTERISTICS (25 °C, unless otherwise noted)

Fig. 1 - Typical Output Characteristics, $T_C = 25^\circ\text{C}$

Fig. 3 - Typical Transfer Characteristics

Fig. 2 - Typical Output Characteristics, $T_C = 175^\circ\text{C}$

Fig. 4 - Normalized On-Resistance vs. Temperature

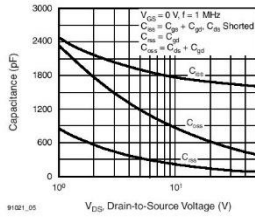


Fig. 5 - Typical Capacitance vs. Drain-to-Source Voltage

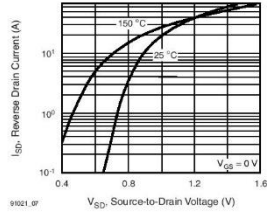


Fig. 7 - Typical Source-Drain Diode Forward Voltage

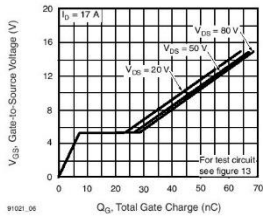


Fig. 6 - Typical Gate Charge vs. Gate-to-Source Voltage

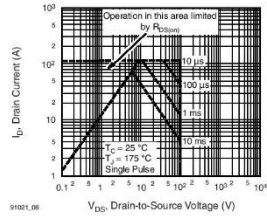
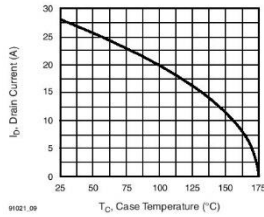
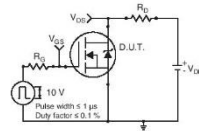
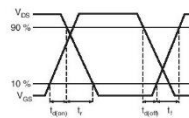
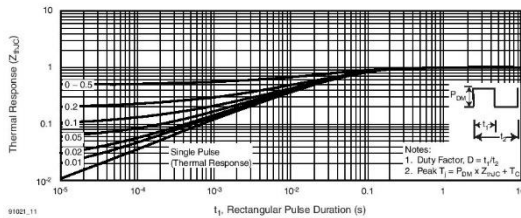


Fig. 8 - Maximum Safe Operating Area


Fig. 9 - Maximum Drain Current vs. Case Temperature

Fig. 10a - Switching Time Test Circuit

Fig. 10b - Switching Time Waveforms

Fig. 11 - Maximum Effective Transient Thermal Impedance, Junction-to-Case

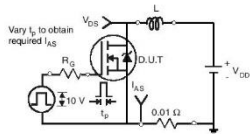


Fig. 12a - Unclamped Inductive Test Circuit

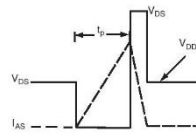


Fig. 12b - Unclamped Inductive Waveforms

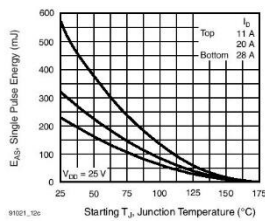


Fig. 12c - Maximum Avalanche Energy vs. Drain Current

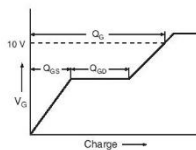


Fig. 13a - Basic Gate Charge Waveform

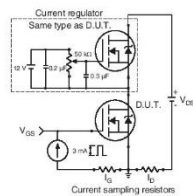
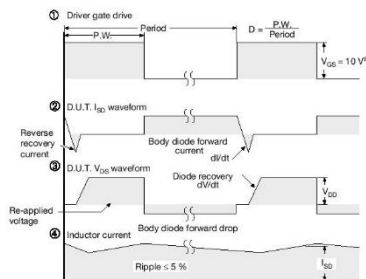
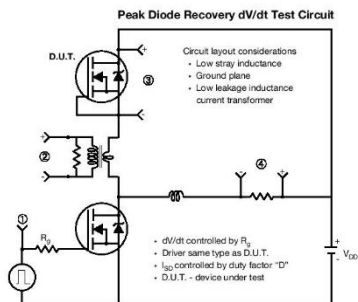


Fig. 13b - Gate Charge Test Circuit



Note
a. $V_{GS} = 5$ V for logic level devices

Fig. 14 - For N-Channel

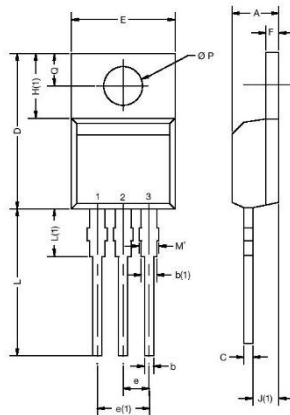
Vishay Siliconix maintains worldwide manufacturing capability. Products may be manufactured at one of several qualified locations. Reliability data for Silicon Technology and Package Reliability represent a composite of all qualified locations. For related documents such as package/tape drawings, part marking, and reliability data, see <http://www.vishay.com/doc?91002>.

Document Number: 91021
S11-0510-Rev. B, 21-Mar-11

www.vishay.com
7

This datasheet is subject to change without notice.
THE PRODUCT DESCRIBED HEREIN AND THIS DATASHEET ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc?91000

TO-220-1

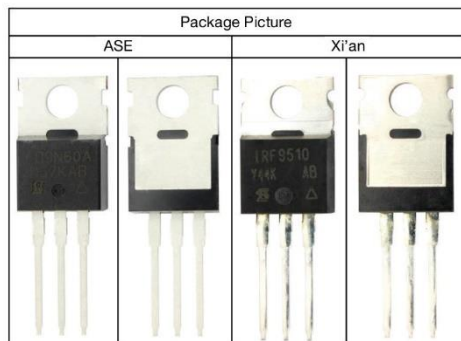


DIM.	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	4.24	4.65	0.167	0.183
b	0.69	1.02	0.027	0.040
b(1)	1.14	1.78	0.045	0.070
c	0.36	0.61	0.014	0.024
D	14.33	15.85	0.564	0.624
E	9.96	10.52	0.392	0.414
e	2.41	2.67	0.095	0.105
e(1)	4.88	5.28	0.192	0.208
F	1.14	1.40	0.045	0.055
H(1)	6.10	6.71	0.240	0.264
J(1)	2.41	2.92	0.095	0.115
L	13.36	14.40	0.526	0.567
L(1)	3.33	4.04	0.131	0.159
$\varnothing P$	3.53	3.94	0.139	0.155
Q	2.54	3.00	0.100	0.118

ECN: X15-0364-Rev. C, 14-Dec-15
DWG: 6031

Note

- M' = 0.052 inches to 0.064 inches (dimension including protrusion), heatsink hole for HVM



Revision: 14-Dec-15

1

Document Number: 66542

For technical questions, contact: hvm@vishay.com

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc391000

3. Datasheet Optocoupler PC817

SHARP

PC817X Series

■ Absolute Maximum Ratings (T_a=25°C)

	Parameter	Symbol	Rating	Unit
Input	Forward current	I _F	50	mA
	^{*1} Peak forward current	I _{FM}	1	A
	Reverse voltage	V _R	6	V
	Power dissipation	P	70	mW
Output	Collector-emitter voltage	V _{CEO}	^{*4} 80	V
	Emitter-collector voltage	V _{ECO}	6	V
	Collector current	I _C	50	mA
	Collector power dissipation	P _C	150	mW
	Total power dissipation	P _{tot}	200	mW
	^{*2} Isolation voltage	V _{iso (rms)}	5.0	kV
	Operating temperature	T _{op}	-30 to +100	°C
	Storage temperature	T _{stg}	-55 to +125	°C
	^{*3} Soldering temperature	T _{sld}	260	°C

^{*1} Pulse width≤100μs, Duty ratio: 0.001

^{*2} 40 to 60%RH, AC for 1minute, f=60Hz

^{*3} For 10s

^{*4} Up to Date code "P7" (July 2002) V_{CEO}: 35V.

■ Electro-optical Characteristics (T_a=25°C)

	Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input	Forward voltage	V _F	I _F =20mA	—	1.2	1.4	V
	Peak forward voltage	V _{FM}	I _{FM} =0.5A	—	—	3.0	V
	Reverse current	I _R	V _R =4V	—	—	10	μA
	Terminal capacitance	C _t	V=0, f=1kHz	—	30	250	pF
Output	Collector dark current	I _{CEO}	V _{CE} =50V, I _F =0	—	—	100	nA
	Collector-emitter breakdown voltage	BV _{CEO}	I _C =0.1mA, I _F =0	^{*5} 80	—	—	V
	Emitter-collector breakdown voltage	BV _{ECO}	I _E =10μA, I _F =0	6	—	—	V
	Collector current	I _C	I _F =5mA, V _{CE} =5V	2.5	—	30.0	mA
	Collector-emitter saturation voltage	V _{CE(sat)}	I _F =20mA, I _C =1mA	—	0.1	0.2	V
Transfer characteristics	Isolation resistance	R _{iso}	DC500V, 40 to 60%RH	5×10 ¹⁰	1×10 ¹¹	—	Ω
	Floating capacitance	C _f	V=0, f=1MHz	—	0.6	1.0	pF
	Cut-off frequency	f _c	V _{CE} =5V, I _C =2mA, R _L =100Ω, -3dB	—	80	—	kHz
	Response time	Rise time	V _{CE} =2V, I _C =2mA, R _L =100Ω	—	4	18	μs
		Fall time		—	3	18	μs

^{*5} From the production Date code "J5" (May 1997) to "P7" (July 2002), however the products were screened by BV_{CEO}≥70V.

Sheet No.: D2-A03101EN

PC817X Series

DIP 4pin General Purpose Photocoupler

*4-channel package type is also available.
(model No. **PC847X Series**)



■ Description

PC817X Series contains an IRED optically coupled to a phototransistor.

It is packaged in a 4pin DIP, available in wide-lead spacing option and SMT gullwing lead-form option.

Input-output isolation voltage(rms) is 5.0kV.

Collector-emitter voltage is 80V^(*) and CTR is 50% to 600% at input current of 5mA.

■ Features

1. 4pin DIP package
2. Double transfer mold package (Ideal for Flow Soldering)
3. High collector-emitter voltage (V_{CEO} :80V^(*))
4. Current transfer ratio (CTR : MIN. 50% at I_F =5 mA, V_{CE} =5V)
5. Several CTR ranks available
6. High isolation voltage between input and output ($V_{iso(rms)}$: 5.0 kV)

(*) Up to Date code "P7" (July 2002) V_{CEO} : 35V.
From the production Date code "J5" (May 1997) to "P7" (July 2002), however the products were screened by BV_{CEO} ≥70V.

■ Agency approvals/Compliance

1. Recognized by UL1577 (Double protection isolation), file No. E64380 (as model No. **PC817**)
2. Package resin : UL flammability grade (94V-0)

■ Applications

1. I/O isolation for MCUs (Micro Controller Units)
2. Noise suppression in switching circuits
3. Signal transmission between circuits of different potentials and impedances

Notice: The content of data sheet is subject to change without prior notice.
In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

4. Datasheet Bluetooth HC-05



Tech Support: info@iteadstudio.com

HC-05

-Bluetooth to Serial Port Module

Overview



HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup.

Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore O4-External single chip Bluetooth system with CMOS technology and with AFH(Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm. Hope it will simplify your overall design/development cycle.

Specifications

Hardware features

- Typical -80dBm sensitivity
- Up to +4dBm RF transmit power
- Low Power 1.8V Operation ,1.8 to 3.6V I/O
- PIO control
- UART interface with programmable baud rate
- With integrated antenna
- With edge connector

HC-05 Bluetooth module

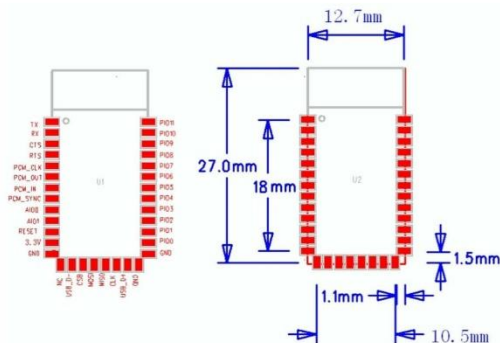
iteadstudio.com

06.18.2010

Software features

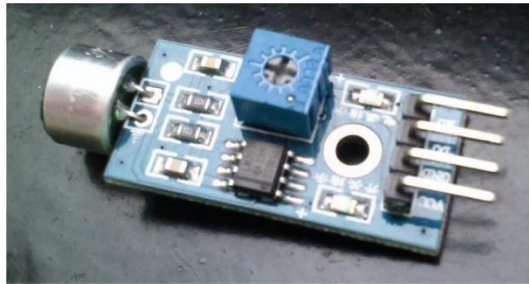
- Default Baud rate: 38400, Data bits:8, Stop bit:1,Parity:No parity, Data control: has. Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.
- Given a rising pulse in PIO0, device will be disconnected.
- Status instruction port PIO1: low-disconnected, high-connected;
- PIO10 and PIO11 can be connected to red and blue led separately. When master and slave are paired, red and blue led blinks 1time/2s in interval, while disconnected only blue led blinks 2times/s.
- Auto-connect to the last device on power as default.
- Permit pairing device to connect as default.
- Auto-pairing PINCODE:"0000" as default
- Auto-reconnect in 30 min when disconnected as a result of beyond the range of connection.

Hardware



5. *Datasheet* Sensor Suara KY-038

SOUND SENSOR MODULE



Description

The sound sensor module provides an easy way to detect sound and is generally used for detecting sound intensity. This module can be used for security, switch, and monitoring applications. Its accuracy can be easily adjusted for the convenience of usage.

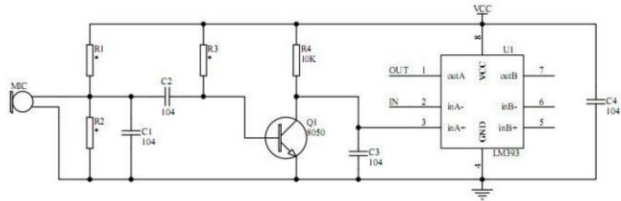
It uses a microphone which supplies the input to an amplifier, peak detector and buffer. When the sensor detects a sound, it processes an output signal voltage which is sent to a microcontroller then performs necessary processing.

Specifications

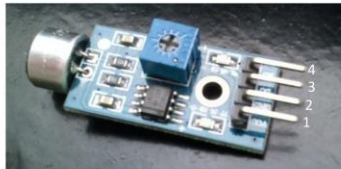
- Operating voltage 3.3V-5V
- Output model: digital switch outputs (0 and 1, high or low level)
- With a mounting screw hole

- PCB size: 3.4cm * 1.6cm

Schematic Diagram



Pin Configuration



1. VCC: 3.3V-5V DC
2. GND: ground
3. DO: digital output
4. AO: analog output

6. Datasheet Raspberry Pi 3

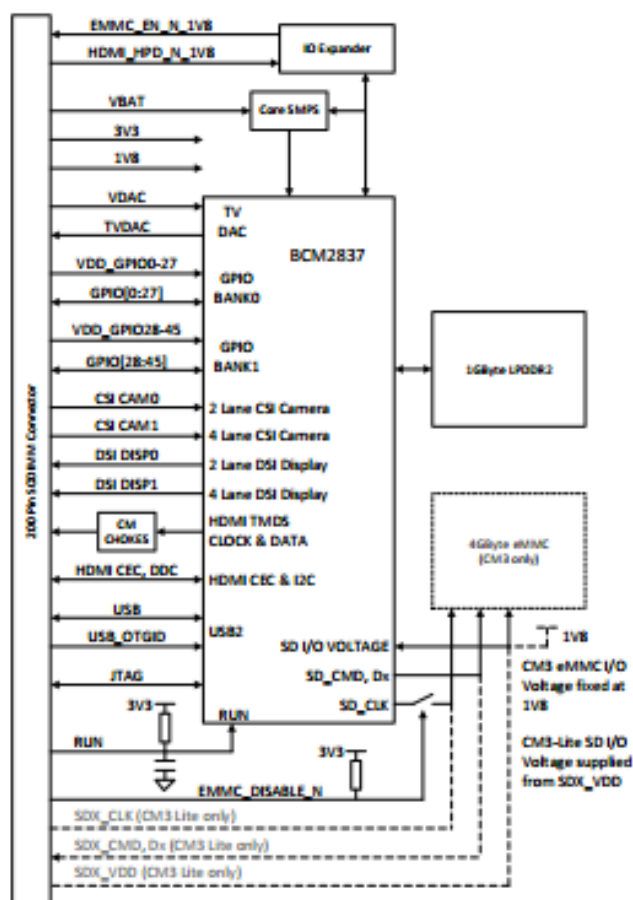


Figure 2: CM3/CM3L Block Diagram

Pin Name	DIR	Voltage Ref	PEN* State	If Unused	Description/Notes
RUN and Host Control (see test for usage guide)					
RUN	I	3V3 ^b	Pull High	Leave open	Has internal 10k pull up
EMMC_DISABLE_N	I	3V3 ^b	Pull High	Leave open	Has internal 10k pull up
EMMC_EN_N, IV8	O	1V8	Pull High	Leave open	Has internal 2k2 pull up
GPIO					
GPIO[27:30]	I/O	GPIO0-27, VDD	Pull or Hi-Z ^c	Leave open	GPIO Bank 0
GPIO[45:28]	I/O	GPIO28-45, VDD	Pull or Hi-Z ^c	Leave open	GPIO Bank 1
Primary SD Interface^{d,e}					
SDX_CLK	O	SDX, VDD	Pull High	Leave open	Primary SD interface CLK
SDX_CMD	I/O	SDX, VDD	Pull High	Leave open	Primary SD interface CMD
SDX_Dx	I/O	SDX, VDD	Pull High	Leave open	Primary SD interface DATA
USB Interface					
USB_Dx	I/O	-	Z	Leave open	Serial interface
USB_OTGID	I	3V3		Tie to GND	OTG pin detect
HDMI Interface					
HDMI_SCL	I/O	3V3 ^b	Z/ ^f	Leave open	DDC Clock (5.5V tolerant)
HDMI_SDA	I/O	3V3 ^b	Z/ ^f	Leave open	DDC Data (5.5V tolerant)
HDMI_CDC	I/O	3V3	Z	Leave open	CDC (has internal 27k pull up)
HDMI_CLKx	O	-	Z	Leave open	HDMI serial clock
HDMI_Dx	O	-	Z	Leave open	HDMI serial data
HDMI_HPD_N, IV8	I	1V8	Pull High	Leave open	HDMI hotplug detect
CAM0 (CSI0) 2-lane Interface					
CAM0_Cx	I	-	Z	Leave open	Serial clock
CAM0_Dx	I	-	Z	Leave open	Serial data
CAM1 (CSI1) 4-lane Interface					
CAM1_Cx	I	-	Z	Leave open	Serial clock
CAM1_Dx	I	-	Z	Leave open	Serial data
DSI0 (Display 0) 2-lane Interface					
DSI0_Cx	O	-	Z	Leave open	Serial clock
DSI0_Dx	O	-	Z	Leave open	Serial data
DSI1 (Display 1) 4-lane Interface					
DSI1_Cx	O	-	Z	Leave open	Serial clock
DSI1_Dx	O	-	Z	Leave open	Serial data
TV Out					
TVDAC	O	-	Z	Leave open	Composite video DAC output
JTAG Interface					
TMS	I	3V3	Z	Leave open	Has internal 50k pull up
TRST_N	I	3V3	Z	Leave open	Has internal 50k pull up
TCK	I	3V3	Z	Leave open	Has internal 50k pull up
TDI	I	3V3	Z	Leave open	Has internal 50k pull up
TDO	O	3V3	O	Leave open	Has internal 50k pull up

* The PEN column indicates power-down state (when RUN pin LOW)

^b Must be driven by an open-collector driver

^c GPIO have software enabled pulls which keep state over power-down

^d Only available on Lite variants

^e The CM will always try to boot from this interface first

^f Requires external pull-up resistor to 5V as per HDMI spec

Table 3: Pin Functions

Caution! Stresses above those listed in Table 5 may cause permanent damage to the device. This is a stress rating only; functional operation of the device under these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Symbol	Parameter	Minimum	Maximum	Unit
V _{BAT}	Core SMPS Supply	-0.5	6.0	V
3V3	3V3 Supply Voltage	-0.5	4.10	V
1V8	1V8 Supply Voltage	-0.5	2.10	V
VDAC	TV DAC Supply	-0.5	4.10	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	-0.5	4.10	V
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	-0.5	4.10	V
SDX_VDD	Primary SD/eMMC Supply Voltage	-0.5	4.10	V

Table 4: Absolute Maximum Ratings

DC Characteristics are defined in Table 5

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V _{IL}	Input low voltage ^a	VDD_IO = 1.8V VDD_IO = 2.7V	-	-	0.6 0.8	V
V _{IH}	Input high voltage ^a	VDD_IO = 1.8V VDD_IO = 2.7V	1.0 1.3	-	-	V
I _{IL}	Input leakage current	TA = +85°C	-	-	5	μA
C _{IN}	Input capacitance	-	-	5	-	pF
V _{OL}	Output low voltage ^b	VDD_IO = 1.8V, IOL = -2mA VDD_IO = 2.7V, IOL = -2mA	-	-	0.2 0.15	V
V _{OH}	Output high voltage ^b	VDD_IO = 1.8V, IOH = 2mA VDD_IO = 2.7V, IOH = 2mA	1.6 2.5	-	-	V
I _{OL}	Output low current ^c	VDD_IO = 1.8V, VO = 0.4V VDD_IO = 2.7V, VO = 0.4V	12 17	-	-	mA
I _{OH}	Output high current ^c	VDD_IO = 1.8V, VO = 1.4V VDD_IO = 2.7V, VO = 2.3V	10 16	-	-	mA
R _{PV}	Pullup resistor	-	50	-	65	kΩ
R _{PD}	Pulldown resistor	-	50	-	65	kΩ

^a Hysteresis enabled

^b Default drive strength (8mA)

^c Maximum drive strength (16mA)

Table 5: DC Characteristics

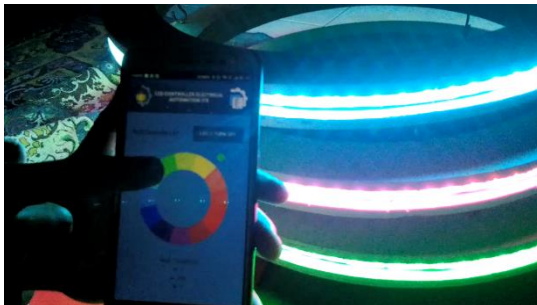
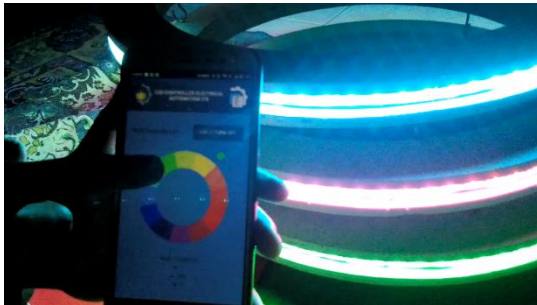
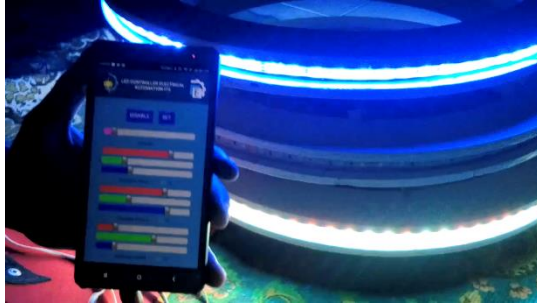
Supply	Description	Minimum	Typical	Maximum	Unit
V _{BAT}	Core SMPS Supply	2.5	-	5.0 + 5%	V
3V3	3V3 Supply Voltage	3.3 - 5%	3.3	3.3 + 5%	V
1V8	1V8 Supply Voltage	1.8 - 5%	1.8	1.8 + 5%	V
VDAC	TV DAC Supply ^a	2.5 - 5%	2.8	3.3 + 5%	V
GPIO0-27_VDD	GPIO0-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
GPIO28-45_VDD	GPIO28-27 I/O Supply Voltage	1.8 - 5%	-	3.3 + 5%	V
SDX_VDD	Primary SD/eMMC Supply Voltage	1.8 - 5%	-	3.3 + 5%	V

^a Requires a clean 2.5-2.8V supply if TV DAC is used, else connect to 3V3

Table 7: Power Supply Operating Ranges

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN F DOKUMENTASI PENGUJIAN





DAFTAR RIWAYAT HIDUP



Nama : Anang Ma'ruf
TTL : Surabaya, 2 Agustus 1996
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat : Jl. Mleto Gg. 1 No. 12 C,
Sukolilo, Surabaya
Telp/HP : 081515051422
E-mail : anangkormus@gmail.com

RIWAYAT PENDIDIKAN

1. 2002 – 2008 : SD Negeri Manyar Sabrangan II, Surabaya
2. 2008 – 2011 : SMP Negeri 30, Surabaya
3. 2011 – 2014 : SMA Negeri 20, Surabaya
4. 2014 – 2017 : Departemen Teknik Elektro Otomasi,
Program Studi Komputer Kontrol – Fakultas
Vokasi Institut Teknologi Sepuluh
Nopember (ITS)

PENGALAMAN KERJA

1. Kerja Praktek di bagian *deployment* Telkom Akses, Sidoarjo
2. Guru Ekstra Komputer SDN Manyar Sabrangan I, Surabaya
3. Guru Ekstra Seni Banjari SMP Islam Raden Paku, Surabaya

PENGALAMAN ORGANISASI

1. Sekretaris Pimpinan Anak Cabang Mulyorejo Ikatan Pelajar
Nahdlatul 'Ulama
2. Ketua Karang Taruna RT. 04, RW. VI, Mleto, Surabaya
3. Staff Departemen PSDM UKM Cinta Rebana ITS
4. Kepala Departemen PSDM UKM Cinta Rebana ITS

-----Halaman ini sengaja dikosongkan-----